Matematica e Commodore 64

Czes Kosniowski

EDIZIONE ITALIANA

GRUPPO EDITORIALE JACKSON

Matematica e Commodore 64

Czes Kosniowski

GRUPPO
EDITORIALE
JACKSON
Via Rosellini, 12
20124 Milano

Copyright per l'edizione originale:
Sunshine Books – 1983
Titolo originale:
MATHEMATICS ON THE COMMODORE 64
Copyright per l'edizione italiana:
Gruppo Editoriale Jackson – Febbraio 1985

TRADUZIONE: Eugenio Roncelli
GRAFICA E IMPAGINAZIONE: Cristina De Venezia
SUPERVISIONE TECNICA: Daria Gianni
COPERTINA: Marcello Longhini
FOTOCOMPOSIZIONE: CorpoNove - Bergamo
STAMPA: S.p.A. Alberto Matarelli - Milano

Tutti i diritti sono riservati. Stampato in Italia. Nessuna parte di questo libro può essere riprodotta, memorizzata in sistemi di archivio, o trasmessa in qualsiasi forma o mezzo, elettronico, meccanico, fotocopia, registrazione o altri senza la preventiva autorizzazione scritta dell'editore.

INDICE

INDICE DETTAGLIATO V
Introduzione all'edizione italiana VII
Note ai programmi VIII
Prefazione IX

CAPITOLO 1
Funzioni semplici? 1

CAPITOLO 2
Trigonometria 7

CAPITOLO 3
Trigonometria Terrestre 25

CAPITOLO 4
Potenze 29

CAPITOLO 5
Serie 47

CAPITOLO 6
Basi Numeriche 57

CAPITOLO 7
Giorni e Settimane 67

CAPITOLO 8
Massimo Comun Divisore (MCD) 77

CAPITOLO 9
Numeri Primi 81

CAPITOLO 10
Miscellanea 91

CAPITOLO 11
Matrici 97

CAPITOLO 12
Codici 111

CAPITOLO 13
Casualità! 123

CAPITOLO 14
Dati in evidenza 133

SOMMARIO PER CAPITOLI 143

INDICE DETTAGLIATO

CAPITOLO 1 — **FUNZIONI SEMPLICI?**

Visualizzazione ordinata dei numeri, arrotondamenti, saldi bancari, saldi bancari scoperti, saldi a colori.

CAPITOLO 2 — **TRIGONOMETRIA**

Disegni in scala, funzioni trigonometriche, funzioni inverse, triangoli non rettangoli, rifrazione, riflessione.

CAPITOLO 3 — **TRIGONOMETRIA TERRESTRE**

La Terra, paralleli e meridiani, calcolo delle distanze.

CAPITOLO 4 — **POTENZE**

Radici quadrate, numeri immaginari, equazioni di II grado, soluzione di equazioni generiche, metodo di Newton, funzioni esponenziali, funzioni logaritmiche, radici di funzioni generiche.

CAPITOLO 5 — **SERIE**

Serie aritmetiche, serie geometriche, calcolo degli interessi, 'Lascia o raddoppia', serie di Fibonacci.

CAPITOLO 6 — **BASI NUMERICHE**

Rappresentazione decimale, coefficienti, numeri binari, numeri esadecimali, conversione di base, i numeri nel C 64, numeri minori di 1, virgola mobile.

CAPITOLO 7 — **GIORNI E SETTIMANE**

Calcolo delle date - formula di Zeller, calendario, gestione delle date.

CAPITOLO 8 — **MASSIMO COMUN DIVISORE**

Fattori comuni, massimo comun divisore, algoritmi euclidei.

CAPITOLO 9 — **NUMERI PRIMI**

Numeri primi e composti, test di primalità, Crivello di Eratostene, numeri primi grandi, numeri di Mersenne, test di primalità probabilistica, numeri pseudo primi.

CAPITOLO 10 – **MISCELLANEA**

Terne pitagoriche e potenze in multiprecisione.

CAPITOLO 11 – **MATRICI**

Introduzione alle matrici, somme di matrici e loro uso, prodotto di matrici e loro uso, matrici nulle, matrici identità, inversione di matrici, sistemi di equazioni.

CAPITOLO 12 – **CODICI**

Codici a sostituzione, codici a matrice, codici a chiave pubblica, codifica e decodifica di messaggi.

CAPITOLO 13 – **CASUALITÀ!**

Testa e croce, uomini e dadi, i giochi delle carte, eventi non equiprobabili.

CAPITOLO 14 – **DATI IN EVIDENZA**

Utilizzo di grandi quantità di dati numerici, il valore medio, massimo, minimo e la dispersione, deviazione standard e varianza, intervalli di confidenza.

INTRODUZIONE ALL'EDIZIONE ITALIANA

Succede spesso che appena in possesso di un piccolo calcolatore, quale il Commodore 64, il manuale diventi 'stretto' di fronte alle possibilità di sviluppo dei programmi.

Un mondo nuovo si apre davanti a noi: ma quale é il miglior modo di ottenere un certo risultato? Oppure quale formula matematica utilizzare per l'applicazione che ci interessa?

Questo libro può risolvere molti dei problemi che i principianti, ma non solo loro, si trovano ad affrontare nell'uso del loro calcolatore.

Gli argomenti trattati, infatti, vanno dalla matematica pura alla geometria, ai codici fino all'analisi statistica.

L'impostazione data dall'Autore é di fornire i necessari elementi matematici, con breve cenno alle basi teoriche e un notevole sviluppo di esempi pratici.

Ogni capitolo è corredato di programmi che, oltre a risolvere il singolo problema, danno lo spunto per altre realizzazioni: in particolare essi possono diventare 'sottoprogrammi' per lo sviluppo di altri di maggior peso.

Come nota d'uso, si consideri che negli esempi é stata sempre utilizzata la notazione anglo-sassone per il punto decimale (che è quella usata dai calcolatori); inoltre per migliore comprensione, gli esempi in denaro mantengono l'espressione in 'sterline' (L.).

Altre precisazioni circa la stesura dei programmi sono date nelle 'Note ai programmi'.

NOTE AI PROGRAMMI

Il Commodore 64 usa "caratteri di controllo" per gestire funzioni quali il movimento del cursore ed i colori. Questi caratteri di controllo appaiono normalmente in campo inverso: per esempio un cuore in campo inverso tra le virgolette pulirà lo schermo posizionando il cursore nell'angolo in alto a destra. Per evitare inconvenienti nel "list" dei programmi non sono stati usati caratteri di controllo; al loro posto si è fatto uso degli equivalenti codici (CHR$).

Qui di seguito è riportato un elenco dei più comuni; altri possono essere cercati nell'Appendice F del Manuale d'Uso del Commodore 64.

CHR$(5)	Bianco
CHR$(17)	Cursore giù
CHR$(28)	Rosso
CHR$(30)	Verde
CHR$(31)	Blu
CHR$(147)	Pulisci video e cursore in alto a destra
CHR$(145)	Cursore su
CHR$(154)	Azzurro
CHR$(157)	Cursore a destra
CHR$(158)	Giallo
CHR$(159)	Ciano

Scrivendo i programmi può essere più comodo utilizzare i caratteri di controllo al posto delle istruzioni CHR$.

PREFAZIONE

Questo libro è scritto per tutti quelli che posseggono un Commodore 64 e vorrebbero conoscere qualche cosa di più riguardo a certe tecniche matematiche.

Voi, probabilmente, avete in mente come scrivere un certo programma, ma, forse, non siete del tutto sicuri della matematica necessaria. Vi serve la funzione COS, ABS, o SGN o altro?

Tutte le funzioni matematiche che trovate nel Commodore 64, sono qui descritte ed il loro uso è illustrato nei programmi. Voi potete 'estrarre' questi programmi ed utilizzarli all'interno dei vostri.

Ma questo libro non è solo un'introduzione a queste funzioni matematiche basilari: esso contiene informazioni essenziali e programmi anche su svariati altri argomenti come i codici e la crittografia, i numeri casuali, le serie, la trigonometria, i numeri primi e l'analisi statistica dei dati.

Si possono utilizzare queste informazioni sia in applicazioni serie che in quelle di divertimento.

Un grosso ringraziamento va ad Ann, Kora e Inga per avermi sopportato durante la stesura di questo libro.

Czes Kosniowski
Newcastle upon Tyne,
SETTEMBRE 1983

CAPITOLO 1

FUNZIONI SEMPLICI?

Visualizzazione ordinata dei numeri

Un numero si dice intero quando non ha parte decimale. Un'ordinata visualizzazione sullo schermo dei numeri interi può essere ottenuta usando in un programma le seguenti linee:

```
L = LEN(STR$(X))
PRINT TAB(25-L)X
```

La funzione STR$(X) converte il numero X in una stringa, LEN ne calcola la lunghezza e TAB muove il cursore nella posizione di schermo richiesta.

```
    9
  123
 -10
   89
```

Per numeri non interi, la visualizzazione non è più corretta.

```
   89
  1.2
-13.89
 .126
```

I numeri sono allineati a destra, ma sarebbe meglio avere allineati i punti decimali: questo può essere ottenuto con l'uso delle funzioni INT(X) e ABS(X).

La funzione INT(X) dà la parte intera di X, cioè il più grande numero intero che è minore o uguale a X. Per esempio

```
INT(1.21) = 1
INT(2) = 2
INT(2.1) = 2
INT(-2) = -2
```

```
INT(-2.1) = -3
INT(9.1) = 9
INT(-9.2) = -10
```

La funzione ABS(X) dà il valore assoluto di X, cioè il numero ignorando il segno + e -. Ad esempio

```
ABS(9.1) = 9.1
ABS(-9.1) = 9.1
```

Il Programma seguente visualizza ordinatamente i numeri decimali sullo schermo.

```
Y = INT(ABS(X)) : L = LEN(STR$(Y))
IF Y = 0 AND X < > 0 THEN L = L-1
PRINT TAB(25-L)X
```

Qui di seguito vi è un tipico esempio di visualizzazione:

```
          3
           .23
        -89.14
  6712399.1
          2.23871
        -1.22
         -.13
```

Nella prima linea del programma, ABS si occupa dei numeri negativi e INT dei numeri non interi. È da notare che la funzione INT da sola non ottiene questo risultato (si veda ad esempio il numero -9.1). La seconda linea si occupa dei numeri compresi tra -1 e 1.

Il suddetto programma illustra un semplice uso delle funzioni INT e ABS. Essa lavora correttamente escluso il caso di numeri molto piccoli (in valore assoluto minore o uguale a 0.01) o molto grandi (in valore assoluto maggiore o uguale a 1000000000). In pratica però, quando appare la notazione scientifica E, la visualizzazione non è più corretta.

```
    3
 -89.14
    1E-04
9.9E+16
```

Potreste aggiungere voi stessi un paio di linee al nostro programma per considerare anche i numeri in notazione scientifica E.

Arrotondamento di numeri

La funzione INTero è utile per "arrotondare" i numeri. Per esempio, se avete in un conto bancario L. 565.58 (sterline) e ricevete un interesse annuo del 9%, dopo un anno vi aspetterete di avere

```
565.58 + 565.58 * 9/100
```

Usando il vostro Commodore 64 potete verificare che il risultato è 616.4822; ma, naturalmente, la banca "arrotonderà" per *difetto* a L. 616.48. Analogamente un importo di 76.6752 sarebbe arrotondato per *eccesso* a 76.68. Il vostro Commodore 64 può fare questo arrotondamento con la seguente linea:

```
X = INT(X*100 + 0.5) /100
```

Dove X è prima moltiplicato per 100 per convertire in pence; poi è aggiunto 0.5 che causa l'arrotondamento per eccesso se la frazione di pence è maggiore o uguale a metà.

La funzione INT ignora la parte decimale ed infine dividendo per 100 riconverte il numero in sterline.

In generale, la linea di programma

```
B = INT(A*10 ↑ D+0.5)/10 ↑ D
```

dà il valore di A arrotondato a D decimali.

Saldi bancari

Il programma che è stato dato precedentemente in questo capitolo per visualizzare ordinatamente i numeri, potrebbe essere usato, ad esempio, in un programma per il saldo bancario. Utilizzandolo (ogni volta che un numero deve essere visualizzato) ci si può attendere una rappresentazione come quella qui sotto.

CAUSALE	DARE	AVERE	SALDO
RIPORTO			596.11
869162	46.22		550.39
869164	169		381.39
869165	15.01		366.38
VERSAM		75.7	442.08

Si noti che quando l'importo comprende 0 pence o un multiplo di 10 pence, lo 0 non significativo non è mostrato. Per forzare la rappresentazione degli 0 non significativi, dobbiamo convertire il numero in una stringa e aggiungere il numero necessario di zeri. Le seguenti linee di programma illustrano come ciò può essere ottenuto.

```
X$ = STR$(X) : L = LEN(STR$(X*100)) : M = LEN(X$)
IF M = L THEN X$ = X$ + "0"
IF M = L-2 THEN X$ = X$ + ".00"
PRINT TAB(25-L)X$
```

Si noti che non si fa uso delle funzioni INT e ABS. Il prospetto bancario precedente apparirà ora così:

CAUSALE	DARE	AVERE	SALDO
RIPORTO			596.61
869162	46.22		550.39
869164	169.00		381.39
869165	15.01		366.38
VERSAM		75.70	442.08

Saldi bancari passivi

Saltuariamente, il saldo bancario può diventare passivo (o andare in rosso). Questo avviene quando il saldo diventa negativo (minore di 0); perciò un saldo di −L. 64.00 significa che siete in passivo di L. 64.00. Le linee di programma che seguono, illustrano l'uso della funzione ABS per stampare saldi bancari e indicare quando l'importo è passivo.

```
X$ = STR$(ABS(X)) : L = LEN(STR$(X*100)): M = LEN(X$)
IF M = L THEN X$ = X$+"0"
IF M = L-2 THEN X$ = X$+".00"
PRINT TAB(25-L)X$;
IF X < 0 THEN PRINT "-";
PRINT
```

Per esempio

CAUSALE	DARE	AVERE	SALDO
RIPORTO			442.08
869166	52.80		389.28
869167	422.00		32.72-

Saldi a colori

La funzione SGN(X) dà il segno (positivo, negativo o zero) del numero X: il risultato è +1 se il numero è positivo, −1 se negativo e 0 se è zero. Per esempio:

```
SGN(9.21) = 1
SGN(−9.1) = −1
SGN(0) = 0
```

Un tipico utilizzo della funzione SGN(X) si ha quando un programma richiede di eseguire differenti sottoprogrammi a seconda che il numero sia positivo, negativo o zero. Ad esempio la linea di programma

```
ON SGN(X) + 2 GOSUB 1000, 1100, 1200
```

obbligherà il programma ad eseguire il sottoprogramma 1000 se X è negativo, il sottoprogramma 1100 se X è zero e il sottoprogramma 1200 se X è positivo.

Un uso interessante della funzione SGN è un semplice metodo per cambiare il colore della visualizzazione. Sul Commodore 64, CHR$(28) indica il rosso, mentre CHR$(30) indica il verde. Perciò CHR$(29+SGN(X)) sarà rosso o verde a seconda che X sia negativo o positivo. Il seguente programma aggiunge questa prestazione a colori al nostro programma di visualizzazione del conto bancario. La prima istruzione POKE è usata per dare un colore di sfondo e rendere meglio visibili il rosso e il verde.

```
POKE 53281,7
X$ = STR$(ABS(X)) : L = LEN(STR$(X*100)) : M = LEN(X$)
IF M = L THEN X$=X$ + "0"
IF M = L−2 THEN X$ = X$+".00"
PRINT CHR$(29+SGN(X))TAB(25−L)X$;
IF X < 0 THENPRINT"−";
PRINT CHR$(30)
```

Questo breve programma visualizza numeri con due decimali (ad esempio sterline e pence) in modo che siano allineati verticalmente. Inoltre il valore assoluto (ABS) di un numero negativo è scritto in rosso seguito dal segno.

CAPITOLO 2

TRIGONOMETRIA

Disegni in scala

Raramente si può misurare direttamente l'altezza di grandi edifici, colline, alberi e simili. Un modo per valutarne l'altezza è di porsi ad una certa distanza da essi: si misura ora l'angolo tra il terreno orizzontale e la punta dell'oggetto (usando un inclinometro e cioè un goniometro perfezionato) e poi la distanza tra voi e l'oggetto. L'altezza può essere valutata facendo un disegno in scala. (Vedi **Figura 1**).

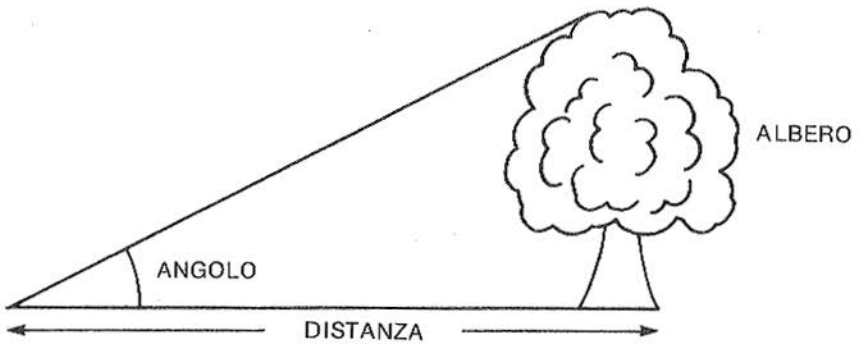

FIGURA 1

Non si può usare la stessa tecnica per misurare l'altezza di un picco montagnoso che è lontano chilometri e coperto dalle nuvole. Le nuvole impediscono la visuale e non si può misurare la distanza orizzontale. Può essere di aiuto uno strumento che usa un radar per individuare la cima della montagna e misura l'angolo e la distanza tra voi e la cima. Ancora una volta, un disegno in scala permette di calcolare l'altezza della montagna. (Vedi **Figura 2**).

Quale ulteriore esempio, supponiamo di voler misurare l'ampiezza di uno stagno o di un lago. (Vedi **Figura 3**).

Anche qui si può fare un disegno in scala con le misure eseguite e calcolare la distanza richiesta.

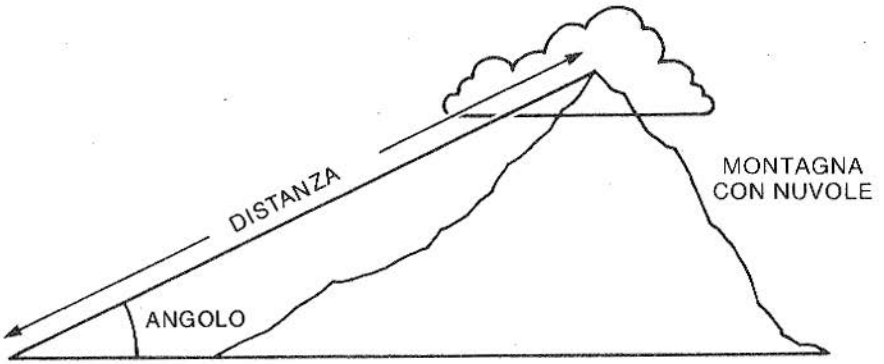

FIGURA 2

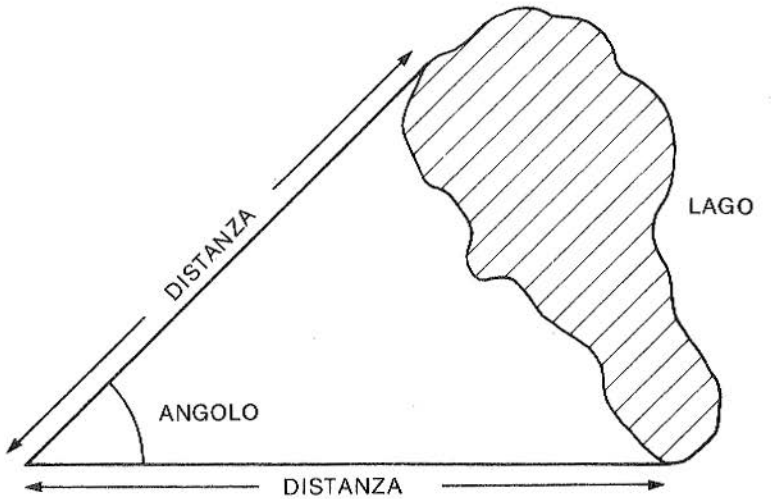

FIGURA 3

Questo è un esempio analogo: un navigatore è in una certa posizione A che si trova esattamente 150 Km a Ovest della città B e 188 Km dalla città C. L'angolo tra le due città visto dalla posizione del navigatore è di 23 gradi. Quanto distano le due città? Ancora una volta il disegno in scala ci dà la risposta.

Benché il disegno in scala dia le risposte cercate, questo metodo è improvvisato: non sempre è comodo e preciso realizzare un disegno in scala. Un'alternativa è di sfruttare la trigonometria usando il vostro Commodore 64.

Le funzioni trigonometriche

Le tre principali funzioni trigonometriche sono SIN (funzione seno), COS (funzione coseno) e TAN (funzione tangente) ed ognuna di esse rappresenta il rapporto tra i vari lati di un triangolo rettangolo. Ad esempio, nella figura sottostante, è mostrato un triangolo rettangolo: l'angolo a sinistra è segnato con X. I tre lati del triangolo sono indicati come il lato adiacente a X, il lato opposto a X e l'ipotenusa (il lato più lungo).

TAN(X) = opposto/adiacente

SIN(X) = opposto/ipotenusa

COS(X) = adiacente/ipotenusa

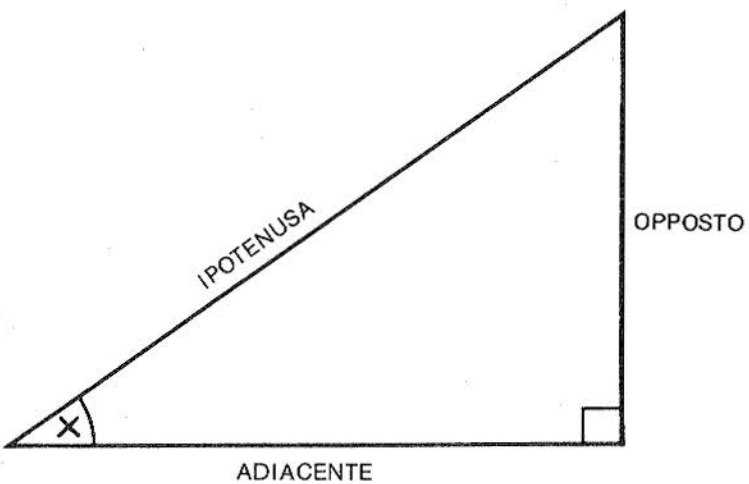

FIGURA 4

Alcuni valori utili da ricordare sono i seguenti:

SIN(0°) = 0
SIN(30°) = 0.5
SIN(45°) = 1/SQR(2)
SIN(60°) = SQR (3)/2
SIN(90°) = 1
COS(0°) = 1
COS(30°) = SQR(3)/2
COS(45°) = 1/SQR(2)
COS(60°) = 1/2
COS(90°) = 0
TAN(0°) = 0
TAN(30°) = 1/SQR(3)
TAN(45°) = 1
TAN(60°) = SQR(3)

Se si conoscono l'angolo X e uno dei tre lati di un triangolo rettangolo, si possono trovare gli altri due lati. Ad esempio, conoscendo l'angolo X e la lunghezza del lato adiacente, gli altri due lati sono dati dalle seguenti formule:

opposto = TAN(X) * adiacente
ipotenusa = adiacente/COS(X)

Un altro modo di descrivere le funzioni trigonometriche è di fare riferimento ad un cerchio di raggio unitario. Si indichi l'angolo come mostrato in **Figura 5**: le varie funzioni trigonometriche sono messe in evidenza.

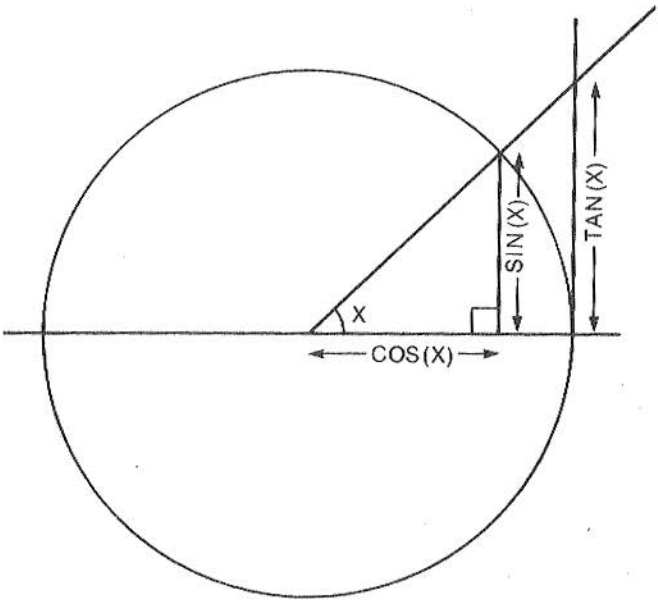

FIGURA 5

In matematica, le distanze sono misurate in orizzontale da sinistra a destra e verticalmente verso l'alto. Questo spiega perché, ad esempio, in **Figura 6**, COS(X) ha un valore negativo.

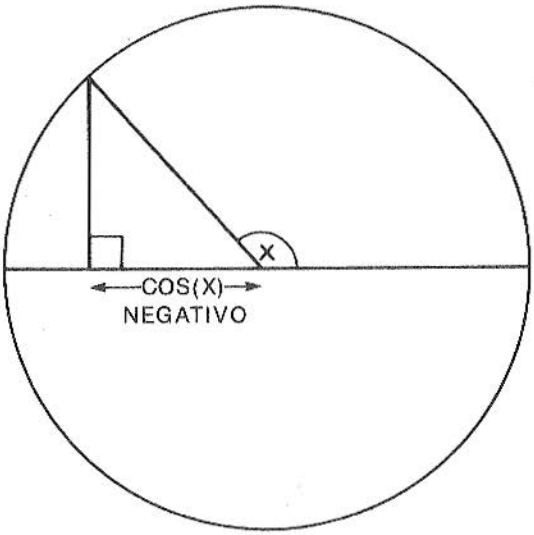

FIGURA 6

Si può ottenere il seno, il coseno e la tangente di un angolo X scrivendo

PRINT SIN(X) etc.

introducendo il valore appropriato di X. L'unico problema è che il Commodore 64, come la maggior parte dei microcomputers, si attende l'angolo in radianti anziché in gradi. Per fortuna, è facile convertire i gradi in radianti e viceversa.

Prima di tutto, che cosa è un radiante? Disegnate un cerchio di raggio unitario; misurate lungo la circonferenza un arco di lunghezza pari al raggio. L'angolo sotteso dall'arco è 1 radiante e vale circa 57 gradi. (Vedi **Figura 7**).

Il numero π è importante e famoso: esso è definito come il rapporto tra la circonferenza del cerchio ed il suo raggio. Il valore (approssimato) di π è memorizzato nel vostro Commodore 64. Semplicemente, digitate

PRINT π

per ottenere il valore memorizzato. In un cerchio di raggio 1, il diametro vale 2 e perciò la circonferenza è 2 π e quindi vi sono 2 π in un cerchio completo. Poiché ci sono 360 gradi in un cerchio completo, si vede che

360 gradi = 2*π radianti e
180 gradi = π radianti

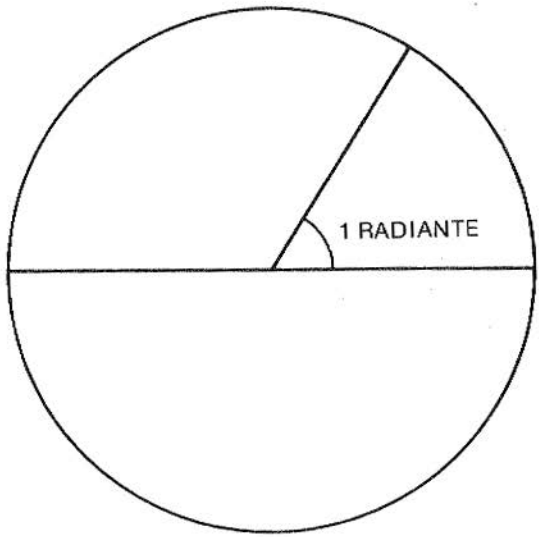

FIGURA 7

Si possono convertire i gradi in radianti e viceversa con facilità, usando le seguenti formule:

X gradi = X * π/180 radianti
Y radianti = Y * 180/π gradi

Il seguente programma può essere usato per trovare i lati di un triangolo rettangolo. È necessario introdurre il valore di un angolo e una lunghezza: il programma calcola gli altri 2 lati.

```
10 REM PROGRAMMA PER TRIANGOLI RETTANGOL
I
20 PRINTCHR$ (147)"          TRIANGOLI RET
TANGOLI"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA VI PERMETTE DI
 TROVARE"
40 PRINT"I LATI DI UN TRIANGOLO RETTANGO
LO"
50 PRINT"NOTI UN LATO E UN ANGOLO"CHR$ (
17)
```

```
60 PRINT"                          #%
                                 # %"
70 PRINT"                        #  %
                                #   %"
80 PRINT"     IPOTENUSA  #       %OPPOSTO
                        #       %"
90 PRINT"                #ANGOLO%
                         ######## "
95 PRINT"                ADIACENTE"CHR$ (17
)
100 REM INPUT DATI
110 INPUT"ANGOLO, IN GRADI";X
120 IFX<=0ORX>=90THENPRINT"ERRORE - NON
E' UN TRIANGOLO":GOTO110
130 PRINTCHR$ (17)"QUALE LATO CONOSCI? 1
 (OPPOSTO)"
140 PRINT"                          2 (ADIACEN
TE)"
145 PRINT"                          3 (IPOTENU
SA)"
150 INPUT"SCRIVI 1, 2 O 3 ";T
160 IFT<1ORT>3ORT<>INT(T)THEN150
170 PRINTCHR$ (17)"SCRIVI LA LUNGHEZZA D
I QUESTO LATO"
180 INPUT"LUNGHEZZA ";L
190 IFL<=0THENPRINT"SCIOCCO - PROVA DI N
UOVO":GOTO180
200 REM CONVERTE IN RADIANTI
205 PI=3.14159265
210 X=X*PI/180
220 REM DIVISIONE DEL LAVORO
230 ONTGOSUB300,350,400
240 PRINTCHR$ (17)"       ECCO FATTO - A
NCORA S/N?"
250 GETG$:IFG$<>"S"ANDG$<>"N"THEN250
260 IFG$="S"THENRUN
270 PRINTCHR$ (147),"   ARRIVEDERCI":END
300 REM LATO OPPOSTO NOTO
310 PRINTCHR$ (17)"LATO ADIACENTE:"L/TAN
(X)
320 PRINT"     IPOTENUSA:"L/SIN(X)
330 RETURN
350 REM LATO ADIACENTE NOTO
```

```
360 PRINTCHR$ (17)"LATO OPPOSTO:"TAN(X)*
L
370 PRINT"    IPOTENUSA:"L/COS(X)
380 RETURN
400 REM IPOTENUSA NOTA
410 PRINTCHR$ (17)"LATO OPPOSTO  :"SIN(X
)*L
420 PRINT"LATO ADIACENTE:"L*COS(X)
430 RETURN
```

Funzioni inverse

Supponendo di conoscere la lunghezza dei lati di un triangolo, possiamo determinare gli angoli? La risposta è affermativa e per farlo utilizziamo le funzioni trigonometriche inverse. Dato un angolo X, la funzione TAN(X) ci dà un valore, la tangente di X. Viceversa, dato un numero N noi possiamo trovare un angolo la cui tangente è quel numero: tale angolo può essere chiamato la tangente inversa di N ed è normalmente indicato con ATN(N), l'arco tangente di N.

Guardiamo il triangolo di **Figura 8**.

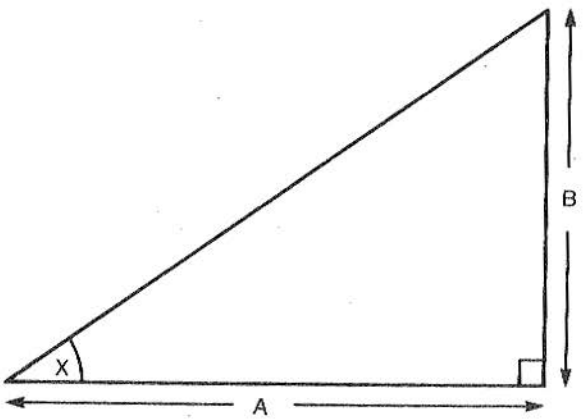

FIGURA 8

Se noi conosciamo il valore di A e B, allora potremo trovare il valore dell'angolo X, perché sappiamo che TAN(X)=B/A e perciò X=ATN(B/A). Si possono introdurre gli opportuni valori in tale espressione e ottenere dal Commodore 64 la risposta. Naturalmente questa sarà in radianti: per averlo in gradi bisogna moltiplicare il risultato per 180/π.

Anche le funzioni SIN e COS hanno la funzione inversa dette rispettivamente ASN (arcoseno) e ACS (arcocoseno). ASN(N) è l'angolo il cui seno è N; analogamente ACS(N) è l'angolo il cui coseno è N. Purtroppo il Commodore 64, come altri microcalcolatori, non contiene queste funzioni nel basic: queste possono comunque essere molto facilmente ottenute dalla funzione ATN.

Per vedere come ottenere ASN da ATN, guardiamo il triangolo rettangolo con ipotenusa di lunghezza 1 della **Figura 9**.

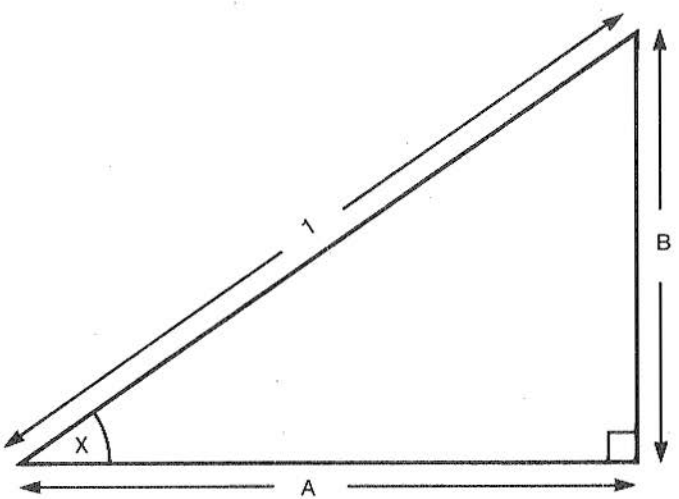

FIGURA 9

Supponiamo di conoscere il valore di B e di voler cercare l'angolo X. Sappiamo che SIN(X)=B e quindi X=ASN(B), ma, come detto, ASN non esiste nel Commodore 64. Se conoscessimo A, potremmo usare ATN poiché è anche X=ATN(B/A). Per trovare A useremo il Teorema di Pitagora.

Rivediamo questo teorema: in parole esso dice che il quadrato dell'ipotenusa di un triangolo rettangolo è uguale alla somma dei quadrati degli altri 2 lati. In simboli abbiamo

$$C^2 = A^2 + B^2$$

dove C è la lunghezza dell'ipotenusa Essendo nel nostro caso uguale a 1, abbiamo:

$$1 = A*A + B*B$$

cioè

$$A*A = 1 - B*B$$

e quindi

$$A = SQR(1-B*B)$$

Poichè X=ATN(B/A) otteniamo

$$X = ATN(B/SQR(1-B*B))$$

ed anche ASN(B) = X, per cui

$$ASN(B) = ATN(B/SQR(1-B*B))$$

Allo stesso modo, si può ottenere una formula per ACS(A), come quella qui sotto

$$ACS(A)=\pi/2 - ATN(A/SQR(1-A*A))$$

Si noti che ACS(A) = π/2 − ASN(A). L'appendice H del Manuale d'Uso del Commodore 64 contiene altri esempi di funzioni matematiche utili, ma che non fanno parte del basic del 64.

Triangoli non rettangoli

I primi due esempi della sezione sui disegni in scala possono essere risolti con il programma Triangoli Rettangoli, ma il terzo esempio (di solito) coinvolge triangoli non rettangoli.

Un triangolo ha tre lati e tre angoli: se noi conosciamo 3 di questi dati (escluso i 3 angoli) possiamo ottenere il valore degli altri 3. Per esempio, potremmo conoscere la lunghezza di 2 lati ed un angolo. Possiamo allora trovare la lunghezza del terzo lato e il valore degli altri due angoli: per farlo usiamo una formula.

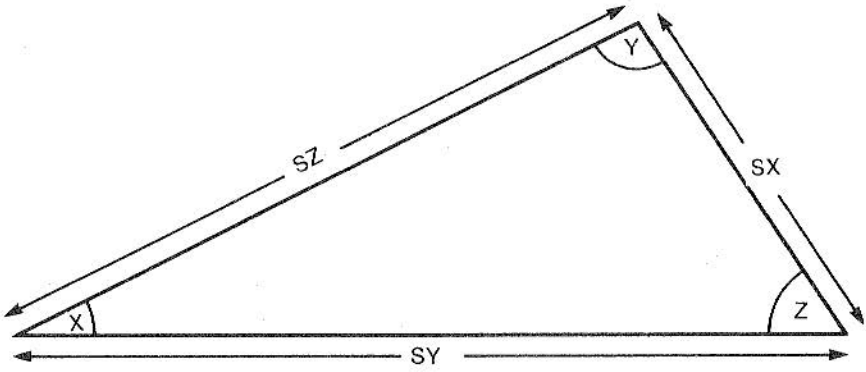

FIGURA 10

Chiamiamo X, Y, Z i tre angoli del nostro triangolo; e SX, SY e SZ i tre lati, essendo SX il lato opposto a X etc. (Vedi **Figura 10**).

Le seguenti formule mettono in relazione i vari lati ed angoli.

La legge dei coseni:

```
SZ*SZ = SX*SX + SY*SY - 2*SX*SY*COS(Z)
SY*SY = SX*SX + SZ*SZ - 2*SX*SZ*COS(Y)
SX*SX = SY*SY + SZ*SZ - 2*SY*SZ*COS(X)
```

La legge dei seni:

```
SIN(X)/SX = SIN(Y)/SY = SIN(Z)/SZ
```

Si noti che se Z è un angolo retto (cioè 90 gradi) allora COS(Z)=0 e la prima formula diventa:

```
SZ*SZ = SX*SX + SY*SY
```

che è proprio il Teorema di Pitagora.

Il programma che segue trova i rimanenti lati ed angoli, supposto che si conosca uno dei seguenti gruppi di dati:

Lato Lato Lato:	Sono noti i 3 lati e si cercano i valori dei 3 angoli.
Lato Lato Angolo:	Sono noti 2 lati ed un angolo che non è tra loro (cioè non è compreso) e si cercano gli altri lati ed angoli.
Lato Angolo Lato:	Sono noti 2 lati e l'angolo tra essi (l'angolo compreso) e si cercano gli altri.

Lato Angolo Angolo: Sono noti 2 angoli ed un lato non tra essi (un lato non compreso) e si cercano gli altri 2 lati ed il terzo angolo.

Angolo Lato Angolo: Sono noti 2 angoli e il lato tra essi (il lato compreso) e si cercano gli altri 2 lati ed il terzo angolo.

Si noti che nel secondo caso (Lato Lato Angolo) sono di solito possibili due triangoli a seconda che l'angolo opposto al terzo lato sia maggiore o minore di 90 gradi. Si veda la **Figura 11** che illustra questo caso.

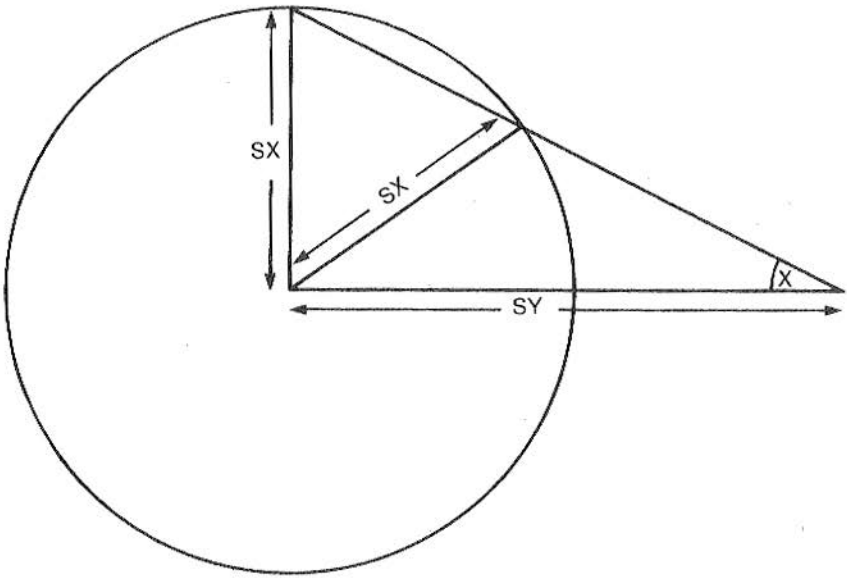

FIGURA 11

```
10 REM TRIANGOLI
20 PRINTCHR$ (147),"     TRIANGOLI"CHR$ (
17)
30 PRINT"QUESTO PROGRAMMA TROVA I RIMANE
NTI LATI"
40 PRINT"E ANGOLI DI UN TRIANGOLO QUALUN
QUE"CHR$ (17)
50 PRINT"QUALI INFORMAZIONI POSSIEDI?"CH
R$ (17)
```

```
60 PRINT"1> LLL : TUTTI E 3 I LATI"CHR$
(17)
70 PRINT"2> LLA : 2 LATI E L'ANGOLO NON
COMPRESO"CHR$ (17)
80 PRINT"3> LAL : 2 LATI E L'ANGOLO COMP
RESO"CHR$ (17)
90 PRINT"4> LAA : 2 ANGOLI E IL LATO NON
 COMPRESO"
100 PRINT"5> ALA : 2 ANGOLI E IL LATO CO
MPRESO"CHR$ (17)
110 REM SCELTA
120 INPUT"SCRIVI IL NUMERO ";N
130 IFN<1ORN>5ORN<>INT(N)THENPRINT,"PROV
A 1,2,3,4 OPPURE 5":GOTO120
140 REM DEFINIZ. FUNZ. ARCOSENO, IN GRAD
I CON 2 DECIMALI
145 PI=3.14159265
150 DEFFNAS(X)=INT((18000/PI)*ATN(X/SQR(
1-X*X))+ .5)/100
160 REM DIVISIONE DEL LAVORO
170 PRINT:ONNGOSUB310,510,710,910,1010
180 PRINTCHR$ (17),"   ANCORA? S/N"
190 GETG$:IFG$<>"S"ANDG$<>"N"THEN190
200 IFG$="S"THENRUN
210 PRINTCHR$ (147),"   ARRIVEDERCI":END
300 REM TUTTI E 3 I LATI
310 PRINT"   *** TUTTI E 3 I LATI CONOSC
IUTI***"CHR$ (17)
320 M=1:GOSUB1110:SX=S
330 M=2:GOSUB1110:SY=S
340 M=3:GOSUB1110:SZ=S
350 A=(SY*SY+SZ*SZ-SX*SX)/(2*SY*SZ)
360 IFABS(A)>=1THENPRINT"NON E' UN TRIAN
GOLO":RETURN
370 PRINT"L'ANGOLO OPPOSTO AL LATO 1 E'
"90-FNAS(A)
380 A=(SX*SX+SZ*SZ-SY*SY)/(2*SX*SZ)
390 PRINTCHR$ (17)"L'ANGOLO OPPOSTO AL L
ATO 2 E'"90-FNAS(A)
400 A=(SX*SX+SY*SY-SZ*SZ)/(2*SX*SY)
410 PRINTCHR$ (17)"L'ANGOLO OPPOSTO AL L
ATO 3 E'"90-FNAS(A)
420 RETURN
```

```
500 REM 2 LATI E UN ANGOLO NON COMPRESO
510 PRINT" *** 2 LATI E UN ANGOLO NON CO
MPRESO***"
520 PRINT"SCRIVI IL LATO OPPOSTO ALL'ANG
OLO        CONOSCIUTO"CHR$ (17)
530 M=1:GOSUB1110:SX=S:GOSUB1210:AX=A
540 M=2:GOSUB1110:SY=S
550 A=SIN(AX)*SY/SX:IFABS(A)>1ORA=0THENP
RINT,"NON E' UN TRIANGOLO":RETURN
560 PRINT"L'ANGOLO OPPOSTO AL LATO 2 E'
MAGGIORE O MINORE DI 90 GRADI?"
570 INPUT"SCRIVI > O <";A$
580 IFA$<>"<"ANDA$<>">"THEN570
590 AY=FNAS(A):IFA$=">"ANDAY<90THENAY=90
+AY
600 PRINTCHR$ (17)"L'ANGOLO OPPOSTO AL L
ATO 2 E'"AYCHR$ (17)
610 AZ=PI-AX-AY*π/180
620 PRINT"LA LUNGHEZZA DEL LATO 3 E'"INT
(SX*SIN(AZ)/SIN(AX)*100)/100CHR$ (17)
630 PRINT"L'ANGOLO OPPOSTO AL LATO 3 E'"
INT((18000/PI)*AZ+.5)/100
640 RETURN
700 REM 2 LATI E L'ANGOLO COMPRESO
710 PRINT"   ***2 LATI E L'ANGOLO COMPRE
SO ***"
720 M=1:GOSUB1110:SX=S
730 M=2:GOSUB1110:SY=S
740 M=3:GOSUB1210:AZ=A
750 SZ=SQR(SX*SX+SY*SY-2*SX*SY*COS(AZ))
760 IFSZ=0THENPRINT,"NON E' UN TRIANGOLO
":RETURN
770 PRINT"LA LUNGHEZZA DEL LATO 3 E'"SZC
HR$ (17)
780 A=(SY*SY+SZ*SZ-SX*SX)/(2*SY*SZ)
790 PRINT"L'ANGOLO OPPOSTO AL LATO 1 E'"
90-FNAS(A)
800 A=(SX*SX+SZ*SZ-SY*SY)/(2*SX*SZ)
810 PRINTCHR$ (17)"L'ANGOLO OPPOSTO AL L
ATO 2 E'"90-FNAS(A)
820 RETURN
900 REM 2 ANGOLI E IL LATO NON COMPRESO
```

```
910 PRINT"*** 2 ANGOLI E IL LATO NON COM
PRESO ***"

920 PRINT"SCRIVI PER PRIMO L' ANGOLO OPP
OSTO AL   LATO CONOSCIUTO"CHR$ (17)
930 M=1:GOSUB1210:AX=A:GOSUB1110:SX=S
940 M=2:GOSUB1210:AY=A
950 A=PI-AX-AY:IFA<=0THENPRINT,"NON E' U
N TRIANGOLO":RETURN
960 PRINT"LA LUNGHEZZA DEL LATO 2 E'"INT
(100*SX*SIN(AY)/SIN(AX))/100CHR$ (17)
970 PRINT"L'ANGOLO OPPOSTO AL LATO 3 E'"
INT((18000/PI)*A+0.5)/100CHR$ (17)
980 PRINT"LA LUNGHEZZA DEL LATO 3 E'"INT
(100*SX*SIN(A)/SIN(AX))/100
990 RETURN
1000 REM 2 ANGOLI ED IL LATO COMPRESO
1010 PRINT"  *** 2 ANGOLI ED IL LATO COM
PRESO ***"
1020 M=1:GOSUB1210:AX=A
1030 M=2:GOSUB1210:AY=A
1040 M=3:GOSUB1110:SZ=S
1050 A=PI-AX-AY:IFA<=0THENPRINT,"NON E'
UN TRIANGOLO":RETURN
1060 PRINT"L'ANGOLO OPPOSTO AL LATO 3 E'
"INT((18000/PI)*A+0.5)/100CHR$ (17)
1070 PRINT"LA LUNGHEZZA DEL LATO 1 E'"IN
T(100*SZ*SIN(AX)/SIN(A))/100CHR$ (17)
1080 PRINT"LA LUNGHEZZA DEL LATO 2 E'"IN
T(100*SZ*SIN(AY)/SIN(A))/100CHR$ (17)
1090 RETURN
1100 REM ACCETTA UN LATO
1110 S=0:PRINT"DAI LA LUNGHEZZA DEL LATO
"M":";:INPUTS:PRINT
1120 IFS<=0THENPRINT,"NON E' UN TRIANGOL
O !":GOTO1110
1130 RETURN
1200 REM ACCETTA UN ANGOLO
1210 A=0:PRINT"DAI L'ANGOLO OPPOSTO AL L
ATO"M":";:INPUTA:PRINT
1220 IFA<=0.0010RA>=180THENPRINT,"NON E'
 UN TRIANGOLO !":GOTO1210
1230 A=A*PI/180:RETURN
```

Rifrazione

Gli oggetti spesso appaiono distorti quando sono visti attraverso il vetro o la plastica; l'acqua appare più bassa di quanto non sia realmente. La ragione è la rifrazione: quando un raggio di luce passa da un mezzo (l'aria) ad un altro (il vetro, l'acqua,...) è curvato o rifratto. L'angolo con cui il raggio incontra il vetro è chiamato angolo di rifrazione. Vedere **Figura 12.**

Per un dato materiale esiste una relazione fissa tra l'angolo di incidenza e di rifrazione. Questo è dato dalla legge di Snell che afferma che il rapporto tra i seni dei due angoli è costante per ogni materiale (in aria). Questo rapporto è detto indice di rifrazione.

$$\text{Indice di rifrazione} = \frac{\text{SIN(angolo di incidenza)}}{\text{SIN(angolo di rifrazione)}}$$

Per il vetro, l'indice di rifrazione è circa 1.5, per l'acqua è 1.333 mentre per il diamante è 2.417.

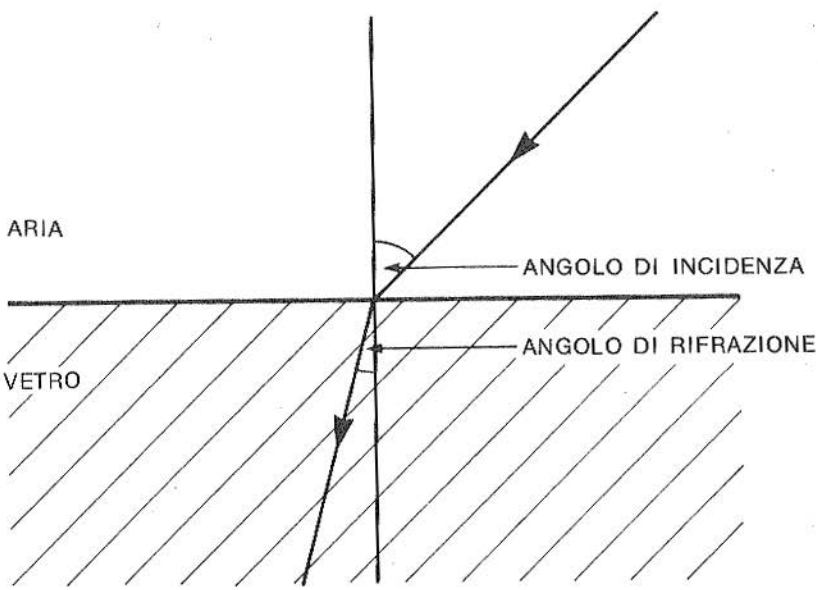

FIGURA 12

Il programma permette di determinare l'angolo di rifrazione, assumendo di conoscere l'angolo di incidenza e l'indice di rifrazione.

```
10 REM RIFRAZIONE
20 PRINTCHR$ (147),"      RIFRAZIONE"CHR$
 (17)
30 PRINT"QUESTO PROGRAMMA CALCOLA L'ANGO
LO DI"
40 PRINT"RIFRAZIONE DI UN RAGGIO DI LUCE
 CHE"
50 PRINT"PASSA DALL'ARIA AD UN ALTRO MEZ
ZO"CHR$ (17)
100 REM INPUT DATI
110 PRINT"SCRIVI L'ANGOLO DI INCIDENZA I
N GRADI"
120 INPUT"ANGOLO: ";X
130 IFX<=0ORX>90THENPRINT"ERRORE - IMPOS
SIBILE":GOTO120
140 PRINTCHR$ (17)"QUALE E' L'INDICE DI
RIFRAZIONE DEL      MEZZO?"
145 INPUT"INDICE DI RIFRAZIONE: ";R
150 IFR<=0THENPRINT"SPIRITOSO - PROVA AN
CORA":GOTO145
160 REM CONVERSIONE IN RADIANTI
170 PI=3.14159265:X=X*PI/180
180 REM CALCOLO
190 Y=SIN(X)/R:Y=Y/SQR(1-Y*Y)
200 PRINTCHR$ (17)"ANGOLO DI RIFRAZIONE:
"INT(10*ATN(Y)*180/PI)/10"GRADI"
210 PRINT"PERCENTUALE DELL'ANGOLO DI INC
IDENZA";INT(ATN(Y)*100/X)
240 PRINTCHR$ (17)"         ECCO FATTO - A
NCORA S/N ?"
250 GETG$:IFG$<>"S"ANDG$<>"N"THEN250
260 IFG$="S"THENRUN
270 PRINTCHR$ (147),"    ARRIVEDERCI":END
```

Riflessione

Un pezzo di vetro o la superficie dell'acqua ogni tanto si comportano come un normale specchio, riflettendo ogni cosa.

Questo avviene quando l'angolo di incidenza è troppo grande ed il raggio di luce è riflesso. Il più piccolo angolo per il quale avviene questo fenomeno è chiamato angolo critico del mezzo ed è calcolato con la seguente formula:

$$\text{SIN(angolo critico)} = \frac{1}{\text{indice di rifrazione}}$$

Perciò l'angolo critico può essere determinato in base all'indice di rifrazione usando la funzione ASN descritta precedentemente.

CAPITOLO 3

TRIGONOMETRIA TERRESTRE

La Terra

La Terra è praticamente una sfera, ma è leggermente schiacciata al polo nord ed al Polo sud. Il raggio all'equatore è di 6378 Km, mentre al Polo è di 6357 Km. La differenza è dello 0,3% circa ed è difficile dire che non è una sfera: il raggio medio è di 6371 Km.

Siamo soliti dire che la minima distanza tra due punti è una 'retta', ma questo vale solo su un piano, o su una qualsiasi superficie 'piana'. Su una sfera, come la Terra, la distanza più breve tra due punti è un arco di cerchio detto cerchio massimo: questo è un cerchio il cui centro passa per il centro della Terra.

I cerchi massimi che passano per i Poli nord e sud sono detti *'linee di longitudine'* e sono associati ad un angolo. La linea di longitudine che passa per Greenwich, Inghilterra, è marcato 0° (0 gradi). Gli altri sono indicati misurando l'angolo al centro della Terra tra la linea di longitudine e quello passante per Greenwich. (Vedere **Figura 13**).

Normalmente le linee di longitudine vanno da 0° a 180° sia ad est che ad ovest.

Le linee di longitudine ci dicono quanto è lontano un punto sulla Terra ad est o ad ovest di Greenwich: per mostrare invece quanto un punto è lontano a nord o a sud dell'Equatore usiamo le linee di latitudine. L'Equatore è detto a latitudine 0° e i cerchi sulla Terra che sono paralleli all'Equatore sono detti *'linee di latitudine'*. L'angolo (misurato al centro della Terra) tra l'Equatore e una linea di latitudine è detto latitudine. (Vedere **Figura 14**).

La latitudine va da 0° a 90° sia a nord che a sud. Il Polo nord è a 90° di latitudine nord ed il Polo sud è a 90° di latitudine sud.

Ogni punto sulla Terra può essere indicato con la sua latitudine e longitudine. Per esempio, Newcastle upon Tyne (Inghilterra) è a 55° di latitudine nord e 1.5° di longitudine ovest: più esattamente i valori sono 54° 58´ N e 1° 36´ O, dove il simbolo ´ indica i minuti essendo un minuto 1/60 di grado.

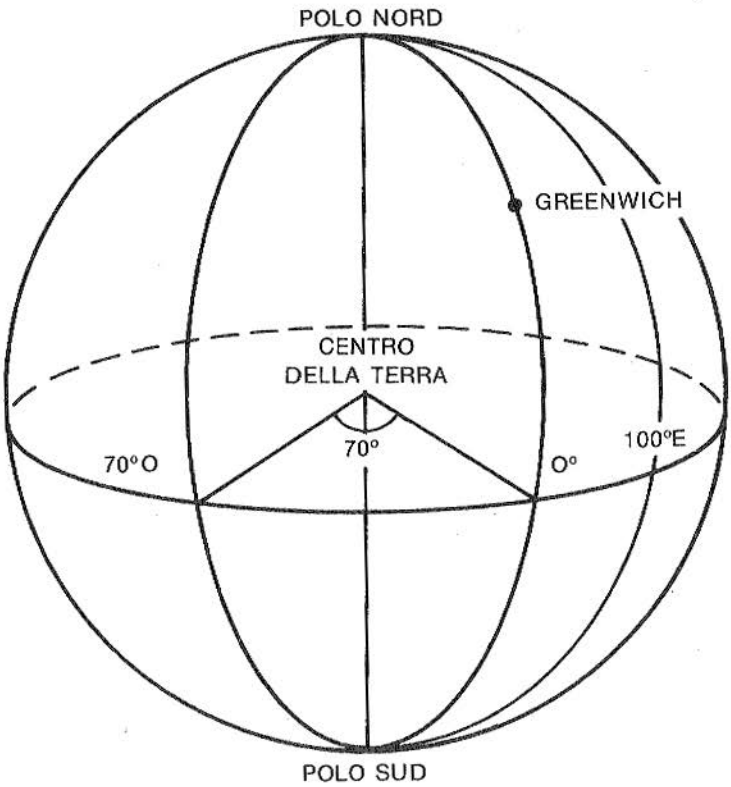

FIGURA 13

Latitudine e longitudine sono una coppia di coordinate sulla superficie della Terra.

Calcolare la distanza (la più breve, sul cerchio massimo) tra due punti non è facile. Per esempio, quale è la distanza tra Newcastle upon Tyne e Parigi in Francia ? (Parigi è a circa 49°N e 2°E). Con un adatto programma sul Commodore 64, ciò non è un problema.

Il programma che segue determina la distanza tra due punti sulla superficie della Terra. I calcoli nel programma sono basati sull'uso ripetuto delle regole del seno e del coseno discusse nel capitolo precedente.

```
10 REM TRIGONOMETRIA TERRESTRE
20 POKE53281,6:PRINTCHR$ (147),CHR$ (154
)"TRIGONOMETRIA TERRESTRE"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA CALCOLA LA DIS
TANZA"
40 PRINT"PIU' CORTA TRA DUE PUNTI SULLA
TERRA"
50 REM INTRODUZIONE DATI
60 FORI=1TO2
70 PRINT:PRINTCHR$ (5)"POSIZIONE";I;CHR$
 (153)CHR$ (17)
80 INPUT"LATITUDINE ";A(I)
90 IFA(I)<0ORA(I)>90THENPRINT"TRA 0 E 90
!!":GOTO80
100 INPUT"   N / S ";A$
110 IFA$<>"N"ANDA$<>"S"THENPRINT"NORD O
SUD!!":GOTO100
120 IFA$="S"THENA(I)=-A(I)
130 PRINT:INPUT"LONGITUDINE ";B(I)
140 IFB(I)<0ORB(I)>180THENPRINT"TRA 0 E
180!!":GOTO130
150 INPUT"   E / O ";A$
160 IFA$<>"E"ANDA$<>"O"THENPRINT"EST O O
VEST(O)!!":GOTO150
170 IFA$="E"THENB(I)=-B(I)
180 NEXT
190 PRINT:PRINTCHR$ (152)"VUOI LA DISTAN
ZA IN MIGLIA O KILOMETRI?"
200 INPUT"M O K";A$:PRINT
210 IFA$<>"M"ANDA$<>"K"THEN200
220 R=6371:B$="KILOMETRI":IFA$="M"THENR=
3960:B$="MIGLIA"
230 PI=3.14159265
240 A1=A(1)*PI/180:A2=A(2)*PI/180:B1=B(1
)*PI/180:B2=B(2)*PI/180
250 B=ABS(B(1)-B(2)):IFB>180THENB=180-B
260 A=ABS(A(1)-A(2))*PI/360:B=B*PI/360
270 X=COS(A1)*SIN(B)*COS(A2)*SIN(B)+SIN(
A)*SIN(A)
280 D=2*R*ATN(SQR(X/(1-X)))
290 PRINTCHR$ (158)"LA DISTANZA E'";INT(
D*100)/100;B$
```

```
300 PRINT:PRINTCHR$ (154),"   ANCORA? S/
N"
310 GETG$:IFG$<>"S"ANDG$<>"N"THEN310
320 IFG$="S"THENRUN
330 PRINTCHR$ (147),"   ARRIVEDERCI":END
```

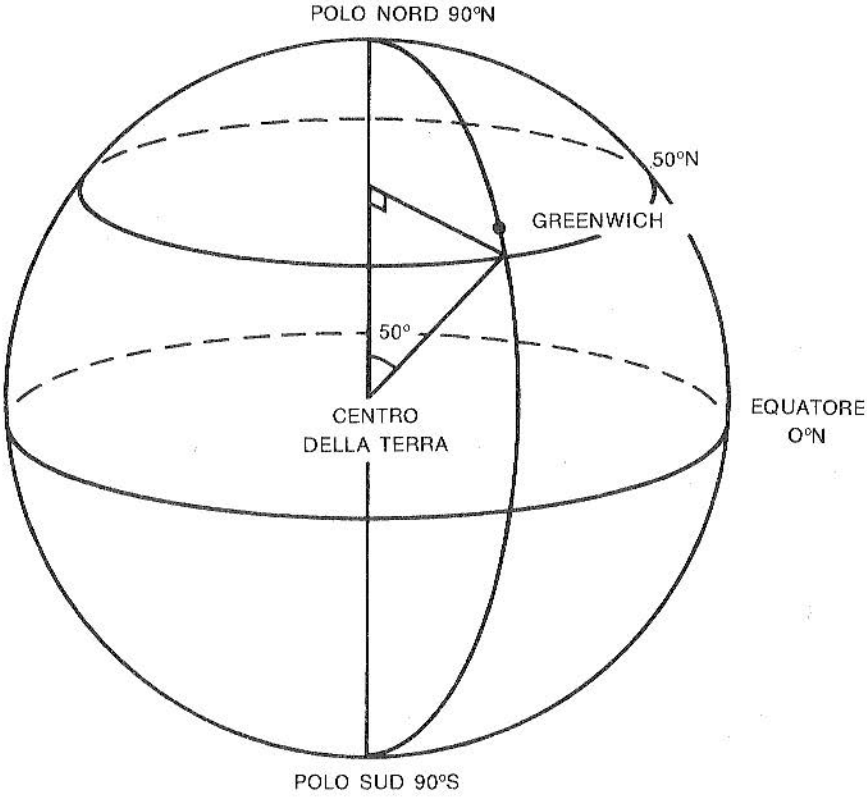

FIGURA 14

CAPITOLO 4

POTENZE

Il quadrato di un numero X è il numero moltiplicato per se stesso, cioè X*X che si scrive X^2 o X↑2 ed è anche detto la seconda potenza di X.

Le potenze di X sono prodotti di un certo numero di X. La prima, seconda, terza, quarta e quinta potenza del numero 2 sono rispettivamente

2, 4, 8, 16 e 32.

Le identifichiamo nell'ordine con

2^1, 2^2, 2^3, 2^4 e 2^5.

Si possono calcolare le potenze di 2 (o di un qualsiasi altro numero) con il seguente semplice programma.

```
10 REM POTENZE
20 PRINTCHR$(147)," POTENZE"CHR$(
17)
30 PRINT"QUESTO PROGRAMMA STAMPA LE POTE
NZE DEL NUMERO INTRODOTTO"CHR$(17)
40 INPUT"NUMERO ";X:PRINT
50 INPUT"QUANTE POTENZE ";N:PRINT
60 IFN<1ORN<>INT(N)THENPRINT"UN INTERO P
ER FAVORE":GOTO50
70 REM CALCOLO E STAMPA
80 PRINTCHR$(5)" I  X↑I"CHR$(17)
90 Y=1
100 FORI=1TON:Y=Y*X:PRINTI;Y:NEXT
110 PRINT:PRINTCHR$(154)," ANCORA? S/
N"
120 GETG$:IFG$<>"S"ANDG$<>"N"THEN120
130 IFG$="S"THENRUN
140 PRINTCHR$(147)," ARRIVEDERCI":END
```

Se il numero N è troppo grande, si può avere un errore di 'overflow' (supero della capacità di calcolo).

Si possono aggiungere alcune linee per verificare un possibile 'overflow' ed uscire dal programma.

```
10 REM POTENZE VERSIONE MIGLIORATA
20 PRINTCHR$ (147),CHR$ (154)"       POTE
NZE"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA TROVA LE POTEN
ZE DEL   NUMERO INTRODOTTO"CHR$ (17)
40 INPUT"NUMERO ";X:PRINT
50 INPUT"QUANTE POTENZE ";N:PRINT
60 IFN<1ORN<>INT(N)THENPRINT"UN INTERO P
ER FAVORE":GOTO50
65 IF(N+1)*LOG(ABS(X))<=126*LOG(2)THEN70
67 PRINT"PROBABILE SUPERO DELLE CAPACITA
' DI CAL-COLO":PRINT
70 REM CALCOLO STAMPA
80 PRINTCHR$ (5)" I  X↑I"CHR$ (17)
90 Y=1
100 FORI=1TON:Y=Y*X:PRINTI;Y:NEXT
110 PRINT:PRINTCHR$ (154),"   ANCORA? S/
N"
120 GETG$:IFG$<>"S"ANDG$<>"N"THEN120
130 IFG$="S"THENRUN
140 PRINTCHR$ (147),"   ARRIVEDERCI":END
```

Il Commodore 64 può anche calcolare la potenza di un numero usando PRINT X↑N.

```
10 REM ANCORA POTENZE
20 PRINTCHR$ (147),CHR$ (154)"   ANCORA
POTENZE"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA TROVA LE POTEN
ZE DEL   NUMERO INTRODOTTO"CHR$ (17)
40 INPUT"NUMERO ";X:PRINT
50 INPUT"QUANTE POTENZE";N:PRINT
60 IFN<1ORN<>INT(N)THENPRINT"UN INTERO P
ER FAVORE":GOTO50
70 IF(N+1)*LOG(ABS(X))<=126*LOG(2)THEN80
```

```
75 PRINT"PROBABILE SUPERO DELLE CAPACITA
' DI CAL-COLO":PRINT
80 REM CALCOLO E STAMPA
90 PRINTCHR$ (5)" I   X↑I"CHR$ (17)
100 FORI=1TON:PRINTI;X↑I:NEXT
110 PRINT:PRINTCHR$ (154),"    ANCORA? S/
N"
120 GETG$:IFG$<>"S"ANDG$<>"N"THEN120
130 IFG$="S"THENRUN
140 PRINTCHR$ (147),"    ARRIVEDERCI":END
```

Come forse saprete o scoprirete presto, N non deve essere necessariamente un numero intero.

Ma che significa 2↑1.7? È ragionevolmente un numero compreso tra 2↑1 e 2↑2, cioè tra 2 e 4, e in effetti il valore è circa 3.24900959. Un metodo approssimato per cercare il valore di 2↑1.7 (senza usare il calcolatore) è quello di usare un grafico. Indicate le potenze di 2 dalla prima alla quinta. Vedi **Figura 15**. Poi disegnate una curva approssimata che passi per quei punti. Vedi **Figura 16**. Su questo grafico si può leggere il valore approssimato di 2↑1.7. Vedi **Figura 17**.

Grafici analoghi possono essere realizzati per ogni altro numero, ma chiaramente non è necessario perché il vostro Commodore 64 vi darà la risposta immediatamente.

Anche le potenze negative di un numero hanno significato e sono semplicement definite dalla seguente regola:

$$X^{-N} = 1/X^{N}$$

Perciò $2^{-2} = 1/2^{2}$ cioè 1/4 o 0.25. Per convenzione la potenza 0 di un numero è 1.

regola:

$$X^{M} * X^{N} = X^{M+N}$$

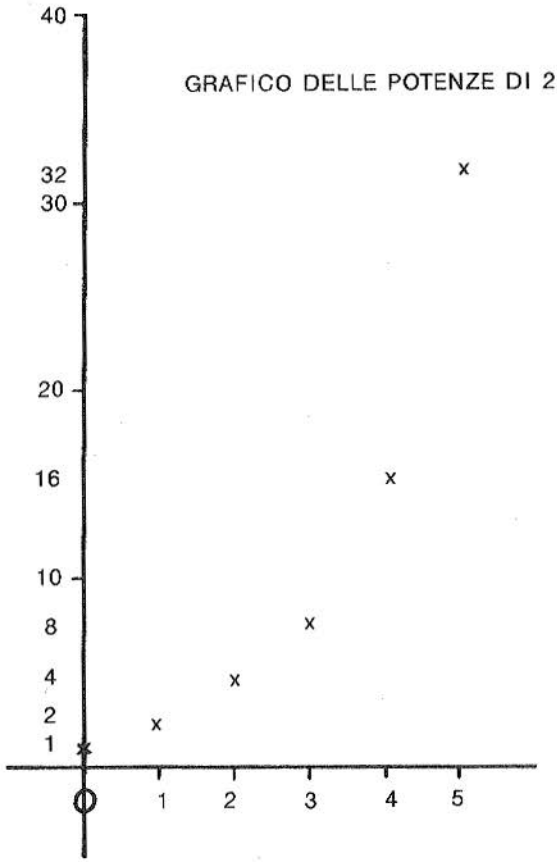

FIGURA 15

Cosicché, ad esempio,

$3^4 * 3^2 = 3^6$

e

$10^{-2} * 10^3 = 10^1$.

Radici quadrate

Per un numero X il numero $X^{1/2}$ ha un nome particolare — la *radice quadrata* di X —. Essa è quel valore il cui quadrato è il numero e perciò la radice quadrata di 9 è 3 e quella di 2 è circa 1.41421356 come si può subito verificare moltiplicando il numero per se stesso.

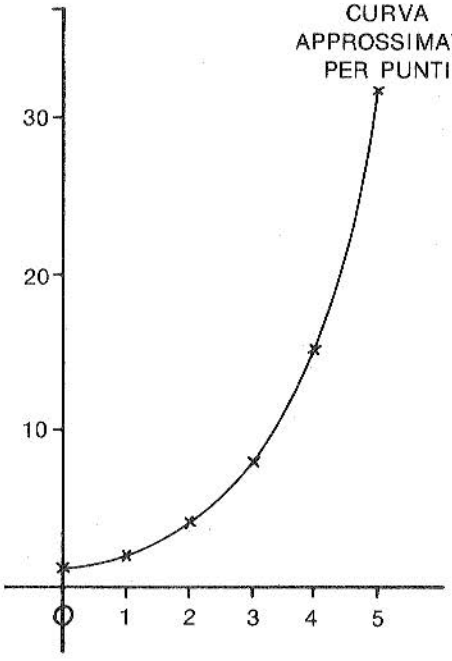

FIGURA 16

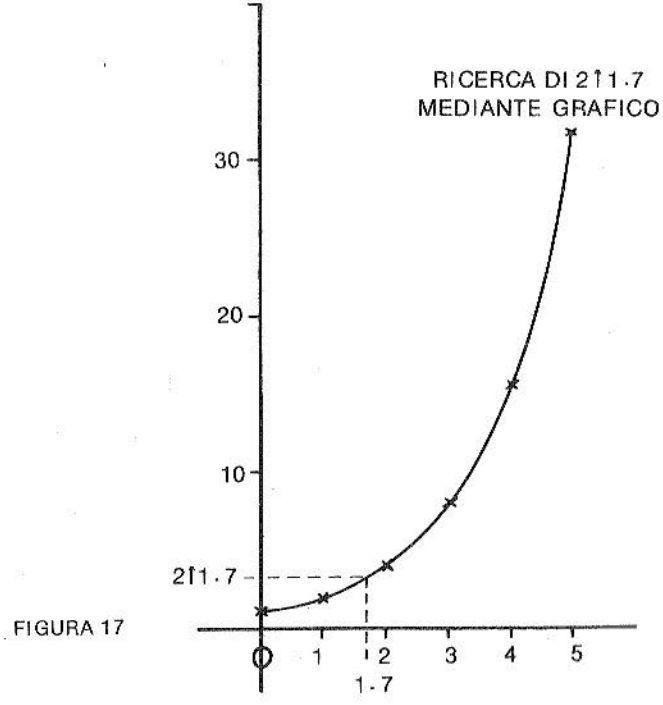

FIGURA 17

La notazione $X^{1/2}$ per la radice quadrata di X, si accorda con il fatto che la moltiplicazione di potenze dà:

$$X^{1/2} * X^{1/2} = X^1$$

che è ovviamente X.

Anche la radice quadrata di X può essere trovata con il Commodore 64 usando PRINT SQR(X).

Numeri immaginari ?

Il quadrato di un numero è sempre positivo e perciò non vi aspettereste di poter trovare la radice quadrata di un numero negativo. Effettivamente se date al Commodore 64 il comando PRINT SQR(–1), vi risponderà con un messaggio ILLEGAL QUANTITY ERROR (errore di quantità illegale).

Può darsi che abbiate sentito parlare di numeri immaginari e di numeri complessi associati con radici quadrate di numeri negativi. I matematici non si sono lasciati spaventare da cose che sembrano impossibili come la radice quadrata di –1. Semplicemente uno di loro ha creato un nuovo simbolo per indicare tale numero. Poniamo quindi che I sia la radice quadrata di –1: non c'è nulla di strano in questo perché I esiste allo stesso modo in cui esistono numeri negativi.

Possiamo sommare il numero I a se stesso, ad altri numeri e moltiplicarlo per un altro numero. Possiamo perciò formare numeri come

2 + 3*I, 1.41412*I, 9–I, 10–8*I.

I numeri che coinvolgono I sono detti numeri complessi, per distinguerli semplicemente dai numeri normali o reali.

Ogni numero complesso può essere scritto nella forma

X + Y * I

per qualsiasi valore reale di X e Y e chiamando X la parte reale e Y la parte immaginaria.

Una volta che abbiamo deciso di usare il simbolo I per SQR(–1), possiamo trovare la radice quadrata di qualsiasi numero negativo.

```
SQR(X) = SQR(ABS(X)*SGN(X))
       = SQR(ABS(X))*SQR(SGN(X))
```

Quindi ad esempio,

$$\begin{aligned} SQR(-9) &= SQR(9)*SQR(-1) \\ &= 3*I \end{aligned}$$

Ma, naturalmente, il Commodore 64 non può lavorare in questo modo perché non conosce nulla dei numeri complessi.

Il programma che segue mostra come inserire i numeri complessi nel vostro Commodore 64 in modo che si possano ottenere semplici operazioni aritmetiche. Notate che la moltiplicazione di numeri complessi segue la seguente regola.

$$\begin{aligned} (A + B*I)*(C + D*I) &= A*C+A*D*I+B*I*C+B*I*D*I \\ &= A*C+A*D*I+B*C*I+B*D*I*I \\ &= A*C+A*D*I+B*C*I+B*D*(-1) \\ &= A*C-B*D+(A*D+B*C)*I \end{aligned}$$

```
10 REM NUMERI COMPLESSI
20 PRINTCHR$ (147),CHR$ (154)" NUMERI CO
MPLESSI"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA ESEGUE PRODOTT
I E DIVI-SIONI";
40 PRINT" DI NUMERI COMPLESSI"CHR$ (17)
50 PRINT"SCRIVI I NUMERI COMPLESSI COME
RICHIESTO"CHR$ (17)
60 PRINTCHR$ (158)"PRIMO NUMERO"CHR$ (17
)
70 INPUT"       PARTE REALE ";A
80 INPUT"PARTE IMMAGINARIA ";B:PRINT
90 PRINT"SECONDO NUMERO"CHR$ (17)
100 INPUT"       PARTE REALE ";C
110 INPUT"PARTE IMMAGINARIA ";D:PRINT
120 PRINTCHR$ (154)"CHE VUOI FARE?"
130 PRINT"1   MOLTIPLICARE I DUE NUMERI"
140 PRINT"2   DIVIDERE IL PRIMO PER IL S
ECONDO "CHR$ (158)
150 INPUT"SCRIVI 1 O 2 ";N:PRINT
160 IFN<>1ANDN<>2THEN150
170 ONNGOSUB300,350
180 PRINT:PRINTCHR$ (154),"   ANCORA? S/
N"
190 GETG$:IFG$<>"S"ANDG$<>"N"THEN190
200 IFG$="S"THENRUN
```

```
210 PRINTCHR$ (147),"    ARRIVEDERCI":END
300 REM MOLTIPLICAZIONE
310 PRINTCHR$ (5)"PRODOTTO DEI NUMERI"CH
R$ (17)
320 PRINT"       PARTE REALE ";A*C-B*D
330 PRINT"PARTE IMMAGINARIA ";A*D+B*C
340 RETURN
350 REM DIVISIONE
360 X=C*C+D*D:IFX=0THENPRINTCHR$ (5)"DIV
ISIONE IMPOSSIBILE":RETURN
370 PRINTCHR$ (5)"DIVISIONE DEL PRIMO PE
R IL SECONDO"CHR$ (17)
380 PRINT"       PARTE REALE ";(A*C+B*D)/
X
390 PRINT"PARTE IMMAGINARIA ";(-A*D+B*C)
/X
400 RETURN
```

Equazioni quadratiche

Nelle soluzioni di molti problemi, prima o poi, ci troviamo di fronte ad equazioni quadratiche, la cui forma generale è:

$$A*X^2 + B*X + C = 0$$

dove A, B, C sono costanti numeriche note con A diverso da 0. Il problema consiste nel trovare quei valori di X che soddisfano l'equazione e che sono dette radici. Ad esempio i valori X=1 e X=2 soddisfano la seguente equazione:

$$X^2-3*X + 2 = 0$$

come si può facilmente verificare.

Ci sono normalmente 2 radici per ogni equazione e spesso sono numeri complessi.

C'è una formula che dà direttamente le radici di un'equazione quadratica; esse sono:

$$\frac{-B + SQR(B^2-4*A*C)}{2*A}$$

e

$$\frac{-B - SQR(B^2-4*A*C)}{2*A}$$

La chiave che indica la natura delle radici si trova in quella parte della formula che è sotto radice,

$SQR(B^2 - 4*A*C)$

che è detta *discriminante*.

se $(B^2 - 4*A*C) > 0$ ci sono 2 radici reali
se $(B^2 - 4*A*C) = 0$ ci sono 2 radici reali uguali
se $(B^2 - 4*A*C) < 0$ ci sono 2 radici complesse

Il programma che segue calcola le radici di un'equazione di II grado.

```
10 REM EQUAZIONI QUADRATICHE
20 PRINTCHR$ (147),CHR$ (154)"EQUAZIONI
QUADRATICHE"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA RISOLVE LE EQU
AZIONI   QUADRATICHE DEL TIPO:"
40 PRINT"A*X*X + B*X +C = 0"CHR$ (17)
50 INPUT"VALORE DI A ";A:PRINT
60 IFA=0THENPRINT"A NON PUO' ESSERE ZERO
: PROVA ANCORA":GOTO50
70 INPUT"VALORE DI B ";B:PRINT
80 INPUT"VALORE DI C ";C:PRINT
90 REM INIZIO CALCOLO
100 D=B*B-4*A*C
110 PRINTCHR$ (158):ONSGN(D)+2GOSUB200,3
00,400
120 PRINT:PRINTCHR$ (154),"   ANCORA? S/
N"
130 GETG$:IFG$<>"S"ANDG$<>"N"THEN130
140 IFG$="S"THENRUN
150 PRINTCHR$ (147),"   ARRIVEDERCI":END
200 REM DUE RADICI COMPLESSE
210 PRINT"CI SONO DUE RADICI COMPLESSE"C
HR$ (17)
220 X=-B/A/2:Y=ABS(SQR(-D)/A/2)
230 PRINTX;"+";Y;CHR$ (157);"*I"
240 PRINTX;"-";Y;CHR$ (157);"*I"
250 RETURN
300 REM DUE RADICI UGUALI
310 PRINT"LE DUE RADICI SONO UGUALI"CHR$
 (17)
320 PRINT-B/A/2
```

```
330 RETURN
400 REM DUE RADICI REALI
410 PRINT"CI SONO DUE RADICI REALI"CHR$
(17)
420 D=SQR(D):X=(-B-D)/2:IFB<0THENX=(-B+D
)/2
430 PRINTX/A:PRINTC/X
440 RETURN
```

Se ci sono 2 radici reali, il programma usa la formula quadratica per trovarne una — la maggiore in valore assoluto. L'altra è calcolata sfruttando il fatto che il prodotto delle 2 radici è C/A.

Soluzione di equazioni di grado maggiore

La soluzione delle equazioni di 2° grado è relativamente semplice: non altrettanto per altre equazioni tipo

$$9*X^5 - 3*X^4 + X^3 - X^2 + 5*X - 4 = 0.$$

Se l'equazione contiene solo potenze di X intere e non negative, come nell'esempio suddetto, essa è detta equazione *polinomiale.* Il più grande valore dell'esponente non nullo di X che contiene l'equazione è detto *grado* del polinomio. Nell'esempio sopra il grado è 5, mentre l'equazione quadratica è di 2° grado ed ha normalmente 2 radici; l'equazione polinomiale di grado N ha N radici.

Salvo casi particolari, non esiste una formula generale per risolvere le equazioni polinominali. Difatti, questa mancanza ha sviluppato alcune interessanti teorie matematiche, ma questi problemi non verrano qui esposti.

Comunque, con il Commodore 64, possiamo cercare di trovare le radici di equazioni polinomiali con un metodo per tentativi. In pratica il calcolatore prova un certo numero di valori per trovare quello che soddisfa l'equazione.

Il programma che segue offre un metodo per trovare le radici reali di un polinomio che è molto rudimentale e talvolta può non trovare le radici, anche se queste esistono. Il programma comunque illustra il metodo generale che ne è alla base.

```
10 REM RADICI DI TPOLINOMI
20 PRINTCHR$ (147),CHR$ (154)"RADICI DI
POLINOMI"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA CALCOLA LE RAD
ICI DI"
```

```
40 PRINT"POLINOMI DEL TIPO"
50 PRINT"A*X↑N+B*X↑(N-1)+...+C*X+D=0"CHR
$ (17)
60 INPUT"GRADO DEL POLINOMIO";N:PRINT
70 IFN<2ORN<>INT(N)THENPRINT"INTERO PER
FAVORE":GOTO60
80 DIMA(N):PRINT"INTRODUCI I COEFFIC. TE
RMINE PER TERMINE"CHR$ (17)
90 FORI=0TON
100 IFI<NTHENPRINT"   COEFFICIENTE DI X↑
";MID$ (STR$(N-I),2);
110 IFI=NTHENPRINT" COEFFICIENTE COSTANT
E";
120 INPUTA(I)
130 IFA(0)=0THENPRINT"NON ZERO PER FAVOR
E":GOTO100
140 NEXT
150 PRINT:PRINT"SCRIVI L'INTERVALLO ENTR
O CUI RICERCARE"
160 PRINT"LE RADICI"CHR$ (17)
170 INPUT"VALORE PIU' BASSO ";A
180 INPUT" VALORE PIU' ALTO ";B:PRINT
190 IFA>=BTHENPRINT"IL PRIMO VALORE DEVE
 ESSERE  PIU' BASSO":GOTO170
200 REM RICERCA DELLE RADICI
210 S=B-A:T=0:TEST=-1:D=1E-9
220 PRINT"PROVA N.";T+1;"**";10↑T;"INTER
VALLI**":GOSUB300
230 IFTESTTHENS=S/10:T=T+1:IFT<4THENGOTO
220
240 IFT=4THENPRINT:PRINT"SPIACENTE - NON
 SI POSSONO TROVARE LE   RADICI"
250 PRINT:PRINTCHR$ (154),"   ANCORA? S/
N"
260 GETG$:IFG$<>"S"ANDG$<>"N"THEN260
270 IFG$="S"THENRUN
280 PRINTCHR$ (147),"   ARRIVEDERCI":END
300 REM RICERCA PASSO PASSO
310 X=A:GOSUB400:X1=A:Y1=Y
320 FORX=A+STOBSTEPS
330 GOSUB400:X2=X:Y2=Y
340 IFY1*Y2<=DTHENGOSUB500
350 Y1=Y2:X1=X
```

```
360 NEXT
370 RETURN
400 REM CALCOLO DEL POLINOMIO
410 Y=A(0):FORI=1TON:Y=Y*X+A(I):NEXT
420 RETURN
500 REM CALCOLO PRECISO DELLA RADICE
510 PRINT"CALCOLO PRECISO DELLA RADICE":
B$=""
520 IFABS(Y1)<DTHENX=X1:GOTO590
530 IFABS(Y2)<DTHENX=X2:GOTO590
540 Z=X:X=(X1+X2)/2:GOSUB400
550 IFABS(Y)<DTHEN590
560 IFABS(Z-X)<DTHENB$="PROBABILE ":GOTO
590
570 IFY*Y2>0THENX2=X:GOTO540
580 X1=X:GOTO540
590 PRINT:PRINT"ESISTE UNA "B$"RADICE: X
=";X:TEST=0:X=B
600 RETURN
```

In qualche caso, il programma dirà che esiste PROBABILMENTE una radice di un certo valore. Per verificare se esiste veramente o controllare quanto è vicina al vero valore della radice, dopo aver risposto N alla richiesta di un'altra prova, scrivete quanto segue:

```
X=Z:GOSUB 400:PRINT Y
```

Il numero stampato vi dice quanto siete vicini al valore di una radice.

Metodo di Newton

Il metodo di ricerca delle radici dato nella sezione precedente può essere migliorato usando il cosiddetto metodo di Newton.

Supponiamo di voler trovare le radici dell'equazione polinomiale

$$9*X^5 - 3*X^4 + X^3 - X^2 + 5*X - 4 = 0$$

Indichiamo con P(X) il polinomio e con P (X) il seguente

$$5*9*X^4 - 4*3*X^3 + 3*X^2 - 2*X + 5.$$

In realtà P´(X) è la *derivata* di P(X), ma questo non ci interessa. Ora, se Y è una radice approssimata di P(X)=0, allora la seguente

$$Y - P(Y)/P'(Y)$$

è un'approssimazione migliore, posto che P´(Y) non sia nullo.

Il seguente programma usa questa tecnica per trovare le radici di un'equazione polinomiale.

```
10 REM RADICI DI POLINOMI - METODO DI NE
WTON
20 PRINTCHR$(147)CHR$(154)" RADICI DI
POLINOMI - METODO DI NEWTON"CHR$(17)
30 PRINT"QUESTO PROGRAMMA RICERCA RADICI
 DI"
40 PRINT"POLINOMI NELLA FORMA"
50 PRINT"A*X↑N + B*X↑(N-1) + ..... + C*X
 + D = 0"CHR$(17)
60 INPUT"GRADO DEL POLINOMIO";N:PRINT
70 IFN<2ORN<>INT(N)THENPRINT"INTERO PER
FAVORE.":GOTO60
80 DIMA(N):PRINT"INTRODUCI I COEFFICIENT
I UN TERMINE ALLA VOLTA"CHR$(17)
90 FORI=0TON
100 IFI<NTHENPRINT"COEFFICIENTE DI X↑";M
ID$(STR$(N-I),2);
110 IFI=NTHENPRINT"COEFFICIENTE COSTANTE
";
120 INPUTA(I)
130 IFA(0)=0THENPRINT"NON ZERO PER FAVOR
E.":GOTO100
140 NEXT
150 PRINT:PRINT"SCRIVI UN PRESUNTO VALOR
E DELLA RADICE"CHR$(17)
160 INPUT"RADICE PRESUNTA";X:PRINT
170 REM CALCOLO DI P´(X)
180 DIMB(N):FORI=0TON:B(I)=(N-1)*A(I):NE
XT
190 GOSUB500:REM RICERCA RADICI
200 REM PARTE FINALE
210 PRINT:PRINTCHR$(154),"   ANCORA? S/
N"
220 GETG$:IFG$<>"S"ANDG$<>"N"THEN220
```

```
230 IFG$="S"THENRUN
240 PRINTCHR$ (147)," ARRIVEDERCI":END
400 REM CALCOLO DEL POLYNOMIO
410 Y=A(0):FORI=1TON:Y=Y*X+A(I):NEXT
420 Y1=B(0):FORI=1TON-1:Y1=Y1*X+B(I):NEX
T
430 RETURN
500 REM RICERCA DELLA SOLUZIONE
510 J=1:D=1E-9:GOSUB400
520 IFABS(Y)<DTHEN590
530 IFY1=0THENPRINT"DIVISIONE PER 0 - DA
I UN'ALTRA RADICE PRESUNTA":RETURN
540 Z=X:X=X-Y/Y1:J=J+1:GOSUB400
550 IFABS(Y)<DTHEN590
560 IFABS(Z-X)<DTHENB$="PROBABILE ":GOTO
590
570 IFJ>1000THENPRINT"SPIACENTE,NON SI P
UO' TROVARE UNA RADICE":RETURN
580 GOTO530
590 PRINT:PRINT"ESISTE UNA "B$"RADICE: X
=";X
600 RETURN
```

Funzioni esponenziali

Le funzioni come 2^X, 10^X sono dette *funzioni esponenziali*, perché la variabile X compare nell'esponente. Le funzioni esponenziali sono calcolabili nel Commodore 64 usando PRINT21X etc. ed esso ne usa una molto importante: la funzione esponenziale EXP(X).

Essa è basata sulle potenze del numero E che vale circa 2.71828183. Non confondete questo numero E con quello che compare con i numeri scritti in notazione scientifica. Il numero E è definito dalla seguente formula:

$$E = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \frac{1}{6!} + \ldots\ldots$$

dove i puntini indicano che la somma è di infiniti termini e il simbolo ! sta per il fattoriale che è definito da

$$N! = N * (N-1) * (N-2) * \ldots * 2 * 1$$

cioè il prodotto degli interi da 1 a N. Per esempio, 4! è 4*3*2*1 uguale a 24.

La funzione esponenziale è definita da

$$EXP(X) = E^X$$
$$= E1X.$$

Cosicché, in particolare E=EXP(1) come potete verificare scrivendo sul Commodore 64

```
E = EXP(1)
PRINT E15,EXP(5)
```

A causa delle proprietà delle potenze, otteniamo le seguenti proprietà delle funzioni esponenziali.

$$EXP(X)*EXP(Y) = EXP(X+Y)$$
$$EXP(X)/EXP(Y) = EXP(X-Y)$$

C'è una formula diretta per calcolare EXP(X) per un qualsiasi numero X ed è la seguente:

$$EXP(X) = 1 + \frac{X}{1!} + \frac{X^2}{2!} + \frac{X^3}{3!} + \frac{X^4}{4!} + \frac{X^5}{5!} + \frac{X^6}{6!} + \ldots.$$

Funzioni logaritmiche

Che numero X soddisfa la relazione EXP(X)=3 ? Poiché EXP(1)=2.718281183, X è poco più di 1 ed in effetti la risposta è circa 1.09861229.

Il numero che soddisfa EXP(X)=N è detto il logaritmo (naturale) di N ed è indicato normalmente con LN(N), ma, come in molti calcolatori, nel Commodore 64 è indicato con LOG(N).

La funzione logaritmica ha le seguenti proprietà:

$$LOG(X*Y)=LOG(X)+LOG(Y)$$
$$LOG(X/Y)=LOG(X)-LOG(Y)$$
$$LOG(X^N)=N*LOG(X)$$

Ed è a causa di queste proprietà che i logaritmi sono importanti nell'esecuzione di moltiplicazioni, divisioni etc. senza calcolatore.

Tra le funzioni logaritmiche ed esponenziali vi sono le seguenti relazioni:

$$EXP(LOG(N)) = N$$
$$LOG(EXP(N)) = N$$

Un semplice uso dei logaritmi è quello di verificare i numeri grandi. Ad esempio il valore X soddisfa la relazione

X<10↑N

se e solo se è soddisfatta la seguente relazione

LOG(X)<N*LOG(10)

Una tale riscrittura della formula è utile perchè il numero 10↑N può causare un ERRORE DI OVERFLOW (DI SUPERO DELLA CAPACITÀ).

Radici di altre funzioni

Data una funzione tipo X*EXP(X)+1, una radice è quel numero che, sostituito nella funzione, dà 0.

Due programmi per la soluzione di equazioni quadratiche sono già stati dati ed il primo di essi può essere adattato per trovare radici di altre funzioni. Quando il programma viene eseguito, vi viene chiesto di introdurre le 2 seguenti linee:

100 DEF FNA(X)= date qui la funzione di X
GOTO 100

Introducete nella prima di queste linee la funzione di cui volete le radici. Se ottenete il messaggio PROBABILE, arrestate il programma e scrivete

PRINT FNA(Z)

per trovare quanto siete vicini alla radice.

```
10 REM RADICI DI FUNZIONI
20 PRINTCHR$ (147),CHR$ (154)"RADICI DI
FUNZIONI"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA TROVA LE RADIC
I DI FUNZIONI"CHR$ (17)
40 PRINT"SCRIVI LA FUNZIONE DI X NELLA S
EGUENTE  FORMA:"
50 PRINTCHR$ (158)"100 DEFFNA(X)=X*EXP(X
)"CHR$ (17)
```

```
60 PRINTCHR$ (154)"POI SCRIVI"CHR$ (17)
70 PRINTCHR$ (158)"GOTO100"CHR$ (154):EN
D
100 DEFFNA(X)=X*EXP(X)*LOG(ABS(X)+1)
150 PRINT:PRINT"SCRIVI L'INTERVALLO ENTR
O CUI ESEGUIRE"
160 PRINT"LA RICERCA DELLE RADICI"CHR$ (
17)
170 INPUT"VALORE PIU' BASSO";A
180 INPUT"VALORE PIU' ALTO ";B:PRINT
190 IFA>=BTHENPRINT"IL PRIMO VALORE DEVE
 ESSERE PIU' BASSO":GOTO170
200 REM RICERCA
210 S=B-A:T=0:TEST=-1:D=1E-9
220 PRINT"PROVA N.";T+1;"**";10↑T;"INTER
VALLI**":GOSUB300
230 IFTESTTHENS=S/10:T=T+1:IFT<4THENGOTO
220
240 IFT=4THENPRINT:PRINT"NON SI PUO' TRO
VARE LA RADICE-SPIACENTE"
250 PRINT:PRINTCHR$ (154),"   ANCORA? S/
N"
260 GETG$:IFG$<>"S"ANDG$<>"N"THEN260
270 IFG$="S"THENRUN
280 PRINTCHR$ (147),"   ARRIVEDERCI":END
300 REM RICERCA PASSO PASSO
310 X=A:Y=FNA(X):X1=A:Y1=Y
320 FORX=A+STOBSTEPS
330 Y=FNA(X):X2=X:Y2=Y
340 IFY1=Y2<=DTHENGOSUB500
350 Y1=Y2:X1=X
360 NEXT
370 RETURN
500 REM RICERCA DELLA SOLUZIONE ESATTA
510 PRINT"     RICERCA DELLA SOLUZIONE E
SATTA":B$=""
520 IFABS(Y1)<DTHENX=X1:GOTO590
530 IFABS(Y2)<DTHENX=X2:GOTO590
540 Z=X:X=(X1+X2)/2:Y=FNA(X)
550 IFABS(Y)<DTHEN590
560 IFABS(Z-X)<DTHENB$="PROBABILE ":GOTO
590
```

```
570 IFY=Y2>0THENX2=X:GOTO540
580 X1=X:GOTO540
590 PRINT:PRINT"ESISTE UNA "B$"RADICE: X
=";X:TEST=0:X=B
600 RETURN
```

CAPITOLO 5
SERIE

Le serie (o sequenze) sono concetti importanti che compaiono ogni momento. Una *serie* è una lista di numeri, ad esempio:

1, 2, 3, 4, 5, 19, 7, 8, 12
20, 18, 16, 14, 12, 10, 8
1, 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001

I singoli numeri di una serie sono detti *termini* e di solito sono ottenuti con un certo metro o criterio, come nelle due precedenti. La seconda è stata creata usando la formula 22–2*N con N da 1 a 7 mentre la terza usa $10/10^N$ con N da 1 a 7. Il vostro Commodore 64 è in grado di generare serie come è illustrato nel semplice programma che segue. Inserite la formula desiderata (in funzione di N) nella seconda riga.

```
10 REM GENERATORE DI SEQUENZE
20 DEF FNA(N) = inserite qui la formula in funzione di N
30 FOR N = 1 TO 10
40 PRINT FNA(N);:IF N<10 THENPRINT CHR$(157)",";
50 NEXT
60 PRINT
```

Qui di seguito, vi sono alcune serie generate con varie formule. Riuscite a scoprire quale formula è stata usata? Controllate la vostra risposta inserendo la formula nel programma precedente. (Le risposte sono date più avanti in questo paragrafo, mentre altri esempi sono dati nei paragrafi successivi).

(a) 1, 6, 11, 16, 21, 26, 31, 36, 41, 46
(b) 1, 2, 4, 8, 16, 32, 64, 128, 256, 512
(c) 1, 4, 9, 16, 25, 36, 49, 64, 81, 100
(d) 1, 2, 4, 7, 11, 16, 22, 29, 37, 46
(e) 4, 4, 8, 12, 20, 32, 52, 84, 136, 220

Le formule per le prime serie non sono molto complicate. Per (a) è 5*N–4, per (b) è 2↑N/2 mentre per (c) è N*N. La quarta non è così facile da indo-

vinare, perché è (N*N–N+2)/2. Infine l'ultima (e) è impossibile indovinarla se non la si è già vista prima, perché essa è:

```
INT(((0.5 + SQR(5)/2)↑N – (0.5–SQR(5)/2)↑N) / SQR(5))
```

Ed ecco altre formule che potrete provare:

```
1+(–1)↑N
N*(–1)↑N
INT(SIN(N)*10)
```

Serie aritmetiche

Una serie *aritmetica*, o progressione aritmetica, è una serie in cui ogni termine è la somma dei precedenti e di una costante. La serie (a) del paragrafo precedente è un esempio di serie aritmetica. La formula generale di tale progressione è data da:

```
A + (N–1)*D
```

nella quale A è il primo termine della serie e D è la differenza comune. Qui di seguito vi sono altri esempi di progressioni aritmetiche.

5,10,15,20,25,30,35,40,45,50	(A=5,D=5)
1,1.5,2,2.5,3,3.5,4,4.5,5,5.5	(A=1,D=0.5)
0,2,4,6,8,10,12,14,16,18	(A=0,D=2)

Il seguente programma vi può aiutare nell'analisi delle serie aritmetiche. Date il primo termine, la differenza comune ed il numero di termini richiesti. Si noti che non è necessaria una formula, perché il Commodore 64 esegue il calcolo ripetitivamente. Inoltre il programma somma tutti i termini della serie e ve ne dà il totale.

```
10 REM SERIE ARITMETICHE
20 PRINTCHR$ (147),"SERIE ARITMETICHE"CH
R$ (17)
30 PRINT" QUESTO PROGRAMMA CREA UNA SERI
E ARITME-TICA"CHR$ (17)
40 INPUT"PRIMO TERMINE ";A:PRINT
50 INPUT"DIFFERENZA COMUNE ";D:PRINT
60 INPUT"NUMERO DEI TERMINI ";N:PRINT
70 IFN<1ORN<>INT(N)THEN60
```

```
80 REM LA SERIE
90 PRINTCHR$ (5)"LA SERIE:"
100 TERM=A:SUM=0
110 FORI=1TON
120 IF38-POS(0)<LEN(STR$(TERM))THENPRINT
130 PRINTTERM;:IFI<NTHENPRINTCHR$ (157)"
,";
140 SUM=SUM+TERM:TERM=TERM+D
150 NEXTI
160 PRINT:PRINT:PRINTCHR$ (159);"LA SOMM
A E' ";SUM
170 PRINT:PRINTCHR$ (154)," ANCORA? S/
N"
180 GETG$:IFG$<>"S"ANDG$<>"N"THEN180
190 IFG$="S"THENRUN
200 PRINTCHR$ (147)," ARRIVEDERCI":END
```

Quale preferite?

Supponete che l'editore di questo libro vi offra un lavoro e vi chieda in che modo preferite essere pagato: 'Quale preferite?' Incominciare con L. 6000 l'anno e aumentare di L. 120 ogni sei mesi oppure partire con L. 6120 e aumentarle di L. 240 ogni anno? Il totale del denaro ricevuto ogni anno con entrambe le scelte segue una progressione aritmetica, ma una scelta è migliore dell'altra - la prima. Volete vedere perché? Osservate i seguenti calcoli:

	I OFFERTA	II OFFERTA
I ANNO	(Primi 6 mesi L. 3000)	
	(Secondi 6 mesi L. 3120)	
	L. 6120)	L. 6120
II ANNO	(Primi 6 mesi L. 3240)	
	(Secondi 6 mesi L. 3360)	
	L. 6600	L. 6360
III ANNO	(Primi 6 mesi L. 3480)	
	(Secondi 6 mesi L. 3600)	
	L. 7080	L. 6600

Potete vedere che se manterrete il lavoro per più di un anno, la prima offerta è migliore. Notate che nella prima offerta, l'importo ricevuto durante il primo semestre di ogni anno aumenta di L. 240 e perciò aumenta di L. 480 ogni anno. Il salario annuale con la prima offerta segue una progressione aritmetica con la formula

6120 + (N–1)*480

mentre per la seconda offerta ha la formula

6120 + (N–1)*240

Supponete che vi dia una terza alternativa: 'Incominciare con L. 5760 all'anno ed aumentare di L. 60 alla fine di ogni trimestre'. Quale scelta preferireste? Speriamo che la risposta sia chiara per voi. Il calcolo procederebbe così:

	III OFFERTA	
I ANNO	(Primi 3 mesi L. 5760/4=	L. 1440)
	(Secondi 3 mesi L. 1500)	
	(Terzi 3 mesi L. 1560)	
	(Quarti 3 mesi L. 1620)	
		L. 6120
II ANNO	(Primi 3 mesi L. 1680)	
	(Secondi 3 mesi L. 1740)	
	(Terzi 3 mesi L. 1800)	
	(Quarti 3 mesi L. 1860)	
		L. 7080

Confrontando ogni trimestre da un anno all'altro l'aumento è di L. 240, ma lo otterrete ogni trimestre e perciò con questa nuova offerta otterrete l'aumento annuo di L. 960.

Serie geometriche

Un altro comune tipo di serie è la serie *geometrica* in cui il rapporto procede in modo costante, ad esempio

2, 6, 18, 54, 162

in cui ogni termine (salvo il primo) è tre volte il precedente. La formula generale per una progressione geometrica è data da

A * RI(N–1)

dove A è il primo termine e R è il rapporto comune.

Qui di seguito vi sono altre serie geometriche.

4,2,1,0.5,0.25,0.125,0.0625	(A=4,R=0.5)
2,–4,8,–16,32,–64,128	(A=2,R=–2)

Il programma seguente vi permette di analizzare serie geometriche: introducete il primo termine, il rapporto comune ed il numero dei termini richiesti: si noti anche qui che non è richiesta una formula perché il Commodore 64 esegue un calcolo ripetitivo. Inoltre il programma somma i termini e ve ne dà il totale.

```
10 REM SERIE GEOMETRICHE
20 PRINTCHR$ (147)," SERIE GEOMETRICHE"C
HR$ (17)
30 PRINT"QUESTO PROGRAMMA CREA SERIE GEO
METRICHE"CHR$ (17)
40 INPUT"PRIMO TERMINE ";A:PRINT
50 INPUT"RAPPORTO COMUNE ";R:PRINT
60 INPUT"NUMERO DEI TERMINI ";N:PRINT
70 IFN<1ORN<>INT(N)THEN60
80 REM LA SERIE
90 PRINTCHR$ (5)" LA SERIE: "CHR$ (17)
100 TERM=A:SUM=0
110 FORI=1TON
120 IF38-POS(0)<LEN(STR$(TERM))THENPRINT
130 PRINTTERM;:IFI<NTHENPRINTCHR$ (157)"
,";
140 SUM=SUM+TERM:TERM=TERM*R
150 NEXTI
160 PRINT:PRINT:PRINTCHR$ (159);"LA SOMM
A E' ";SUM
170 PRINT:PRINTCHR$ (154)," ANCORA? S/
N"
```

```
180 GETG$:IFG$<>"S"ANDG$<>"N"THEN180
190 IFG$="S"THENRUN
200 PRINTCHR$(147),"    ARRIVEDERCI":END
```

Interessi

Il primo di Gennaio, una signora mette in banca L. 100 che le danno il 6% di interesse annuo (alla fine dell'anno). Quale sarà diventata la cifra della signora dopo 10 anni in banca?

Dopo un anno avrà

```
100 + 100 * 6/100
    = 100 + 100 * 0.06
    = 100 * 1.06
```

cioé L. 106. Alla fine di due anni avrà

```
100 * 1.06 + (100*1.06) * 0.06
    = 100 * 1.06 * 1.06
```

e dovrebbe vedere che, dopo 10 anni, avrà

```
100 * 1.06↑10
```

La somma totale alla fine di ogni anno forma una serie geometrica come mostrato qui sotto.

```
100*1.06,100*1.06↑2,100*1.06↑3,100*1.06↑4,
    100*1.06↑5,100*1.06↑6,100*1.06↑7,100*1.06↑8,
        100*1.06↑9,100*1.06↑10
```

Più in generale, se incominciate con un importo A e ricevete un interesse annuo I%, allora dopo N anni, l'importo originale è cresciuto fino al seguente

```
A * (1 + I/100)↑N
```

Interesse giornaliero

Se, con interesse del 6% alla fine dell'anno, sono depositate in banca L. 1000, alla fine di un anno il totale diverrà

```
1000 * 1.06
```

assumendo di non fare ulteriori depositi o prelievi. Se, d'altra parte, la banca pagasse gli interessi ogni 6 mesi (pagando gli interessi sugli interessi cioè l'interesse composto), allora il totale alla fine dell'anno sarebbe

```
1000 * (1.03)12
```

Più in generale, se la banca paga N volta all'anno l'interesse composto del 6%, alla fine di detto anno le L. 1000 sarebbero cresciute al seguente importo:

```
1000 * (1 + 0.06/N)↑N
```

La tabella che segue illustra i differenti importi a seconda della frequenza con cui l'interesse è composto.

N	6% composto	Totale a fine anno (arrotondato)
1	annuale	L. 1060.00
2	semestrale	L. 1060.90
4	trimestrale	L. 1061.36
6	bimensile	L. 1061.52
12	mensile	L. 1061.68
52	settimanale	L. 1061.80
365	giornaliero	L. 1061.83
8760	orario	L. 1061.84

Questa tabella è stata preparata con il Commodore 64 usando questo semplice programmino.

```
10 REM FORMULA PER L'INTERESSE COMPOSTO
20 PRINT "QUANTE VOLTE L'INTERESSE È COMPOSTO"
30 INPUT "NUMERO";N:IF N<>INT(N)THEN30
40 T = 1000*(1+0.06/N)↑N:PRINT INT(T*100)/100
50 PRINT "ANCORA?S/N"
60 GET G$: IF G$<>"S" AND G$<>"N" THEN60
70 IF G$ ="S" THEN RUN
```

Si possono fare modifiche ed aggiunte per permettere altri tassi d'interesse: qui sotto le righe modificate sono indicate con un asterisco.

```
 10 REM FORMULA PER L'INTERESSE COMPOSTO–INTERESSE
    VARIABILE
*15 INPUT "TASSO D'INTERESSE"; I:IF I<=0 OR I>=100 THEN15;
 20 PRINT "INTRODUCI QUANTE VOLTE L'INTERESSE
 È COMPOSTO"
 30 INPUT "NUMERO"; N:IF N<>INT(N) OR N<1 THEN30
*35 IF N>20000 THEN PRINT "ATTENZIONE * LA RISPOSTA PUÒ
 ESSERE IMPRECISA"
 40 T=1000*(1+I/100/N)↑N:PRINT INT(T*100)/100
 50 PRINT "ANCORA?S/N"
 60 GET G$:IFG$<>"S"ANDG$<>"N"THEN60
 70 IFG$="S"THENRUN
```

La linea 35 è stata introdotta per le imprecisioni che si possono avere con numeri molto grandi.

Lascia o raddoppia

Alcuni pensano che non sia obbligatorio perdere sempre quando si gioca d'azzardo.

Per capire questo fatto, vediamo questo gioco: il 'testa o croce': se scommettete 'testa' e compare 'testa' vincete la somma raddoppiata altrimenti perdete la puntata.

Per mostrare che non è obbligatorio perdere, il ragionamento fila così: incominciate con una puntata di L. 1 e, se vincete, chiudete il gioco. Se perdete, giocate ancora con una posta di L. 2. Ogni volta che perdete, raddoppiate la posta. Fermatevi appena vincete e godetevi la somma.

Facciamo l'ipotesi che voi perdiate per i primi quattro lanci e vinciate al quinto: la tabella mostra l'andamento.

	POSTA	PERDITA	VINCITA
Primo lancio	L. 1	L. 1	
Secondo lancio	L. 2	L. 2	
Terzo lancio	L. 4	L. 4	
Quarto lancio	L. 8	L. 8	
Quinto lancio	L. 16		L. 16
TOTALE		L. 15	L. 16
GUADAGNO NETTO =	L. 1		

La serie ottenuta è geometrica: vi è chiaro il criterio per cui non perderete mai?

Serie di Fibonacci

All'inizio del capitolo abbiamo scritto questa serie:

(e) 4,4,8,12,20,32,52,84,136,220

che è data dalla formula

INT(((0.5+SQR(5)/2)↑N −(0.5−SQR(5)/2↑N)/SQR(5))

Piuttosto che usare questa formula, c'è un modo più ovvio per creare questa serie perchè ogni termine, eccetto i primi 2, è la somma dei precedenti.

4 + 4 = 8
4 + 8 = 12
8 + 12 = 20,

e così via.

Serie di numeri così fatte sono dette di Fibonacci. Successe nel 1202 che Leonardo di Pisa, detto Fibonacci, osservasse le serie di numeri associate con le filiazioni di conigli.

Ecco altre due serie di Fibonacci:

2,5,7,12,19,31,50
3,5,8,13,21,34,55

Il programma qui sotto genera serie di Fibonacci fino alla nausea.

```
10 REM SERIE DI FIBONACCI
20 PRINTCHR$ (147),"SERIE DI FIBONACCI"C
HR$ (17)
30 PRINT"QUESTO PROGRAMMA CREA SERIE DI
FIBONACCI"
40 PRINT"SCRIVI DUE NUMERI INTERI, SEPAR
ATI DA   UNA VIRGOLA"CHR$ (17)
50 INPUT"NUMERI ";U,V:PRINT
60 IFU<>INT(U)ORV<>INT(V)THENPRINT"INTER
I PER FAVORE":GOTO50
70 INPUT"QUANTI TERMINI VUOI ";N:PRINT
80 IFN<1ORN<>INT(N)THEN70
90 REM LA SERIE
100 PRINTCHR$ (5)"LA SERIE DI FIBONACCI:
"CHR$ (17)
110 FORI=1TON
120 IF38-POS(0)<LEN(STR$(U))THENPRINT
130 PRINTU;:IFI<NTHENPRINTCHR$ (157)",";
140 W=U+V:U=V:V=W
150 NEXTI
160 PRINT:PRINT:PRINTCHR$ (154)."   ANCO
RA? S/N"
170 GETG$:IFG$<>"S"ANDG$<>"N"THEN170
180 IFG$="S"THENRUN
190 PRINTCHR$ (147)."   ARRIVEDERCI":END
```

Eccovi un esercizio che potrebbe piacervi: usate il vostro Commodore 64 e scrivete un breve programma.

'Scrivete due numeri interi. Formate una serie di Fibonacci sommando coppie di termini per formarne un terzo.

Trovate il rapporto tra ogni termine ed il precedente. Che succede a questo rapporto se il numero dei termini aumenta? Provate il valore di 0.5 + SQR(5)/2'.

CAPITOLO 6
BASI NUMERICHE

Normalmente usiamo i numeri usando la notazione del *sistema decimale*. Ad esempio 1432, che diciamo millequattrocentotrentadue, sta per la più complicata espressione

$$1*10^3 + 4*10^2 + 3*10 + 2.$$

Oppure, in modo leggermente più complesso:

$$1*10^3 + 4*10^2 + 3*10^1 + 2*10^0$$

Poiché $10^1=10$ e $10^0=1$. In altre parole, il numero 1432 è interpretato come somma di multipli di potenze di 10. Gli interi 1,4,3 e 2 sono detti *cifre* del numero essendo 1 la cifra delle migliaia, 4 quella delle centinaia, 3 quella delle decine e 2 quella delle unità. Tecnicamente ci riferiamo ad una rappresentazione dei numeri che è detta *decimale* e diciamo che i numeri sono espressi in *base* 10. Il vocabolo decimale deriva dal latino *decem*, dieci.

Il sistema decimale ha base 10, ma possono essere usate altre basi, in un modo che può essere insieme interessante ed utile. Ad esempio, la rappresentazione in base 2 si è rivelata estremamente importante nei calcolatori e nelle attività che vi sono legate.

Un qualsiasi intero, maggiore di 1, può essere usato come base e qualsiasi numero può essere espresso in qualsiasi base.

Posto che N sia un qualsiasi numero intero e B sia un intero maggiore di 1, per esprimere il numero N in base B, dobbiamo scriverlo nel seguente modo:

$$N = X_m*B^m + X_{m-1}*B^{m-1} \ldots + X_1*B + X_0$$

dove ciascuno dei numeri $X_0, X_1, \ldots, X_m$ sono interi compresi tra 0 e B–1 (controllate che cosa succede sostituendo 10 in B) e sono detti i *coefficienti* del numero N in base B.

Piccoli valori di B, la base, danno lunghe rappresentazioni dei numeri, ma hanno il vantaggio di richiedere poche scelte per i coefficienti. Il caso estremo si ha con B=2, da cui risulta il sistema di numerazione *binario* (dal Latino

binarius, due) in cui solo gli interi 0 e 1 compaiono come coefficienti. Ad esempio

$$86 = 64 + 16 + 4 + 2$$
$$= 1*2^6+0*2^5+1*2^4+0*2^3+1*2^2+1*2+0$$

Perciò il numero 86, espresso in forma binaria è 1010110. I numeri binari sono usati nei calcolatori perché essi sono rappresentati da una catena di zeri e uni ed il motivo sta nel fatto che 0 e 1 possono essere facilmente visti come le due posizioni di un interruttore che può essere aperto o chiuso.

Per basi maggiori di 10, abbiamo bisogno di simboli in più ed il sistema più ovvio è quello di usare le lettere dell'alfabeto A, B, C, etc. Una base comunemente usata è il 16 ed il numero espresso in tale base è detto *esadecimale*. Il vantaggio è che essa richiede pochi coefficienti per rappresentare un numero e questo è facilmente convertibile in numero binario.

Convertire un numero dalla base 10 alla base B è piuttosto semplice. Supponiamo di voler convertire il numero N in base 10 alla base B. Prima di tutto sottraiamo tutti i multipli di B da N.

$$M = INT(N/B) : R_0 = N - B * M$$

Segnatevi il resto chiamandolo R_0 e ripetete il procedimento con M ponendo N=M. Chiamiamo il nuovo resto R_1 e continuiamo così finché M=0. Supponendo che R_s sia l'ultimo resto trovato, allora il numero originale N in base B è

$$R_s .... R_2R_1R_0$$

Vediamo con un esempio concreto. Vogliamo convertire il numero 29 in base 3: i calcoli procedono così:

Passo 1

$N = 29$
$M = INT(29/3)$
$= 9$
$R_0 = 29 - 3*9$
$= 2$

Passo 2

$N = M$
$= 9$
$M = INT(9/3)$
$= 3$
$R_1 = 9 - 3*3$
$= 0$

Passo 3

$N = M$
$= 3$
$M = INT(3/3)$
$= 1$
$R_2 = 3 - 3*1$
$= 0$

Passo 4

$N = M$
$= 1$
$M = INT(1/3)$
$= 0$
$R_3 = 1 - 3*0$
$= 1$

Il procedimento si ferma dopo il quarto passo in cui N vale 0. Il numero 29 in base 3 è 1002.

Il programma che segue converte i numeri da una base all'altra e voi potrete convertire i numeri da base 10 a base 16. etc. Per basi maggiori di 10 sono utilizzate le lettere A, B, C, etc. per indicare i numeri 10, 11, 12 etc.

```
10 REM CONVERSIONE DI BASE
20 PRINTCHR$ (147)," CONVERSIONE DI BASE
"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA CONVERTE NUMER
I INTERI DA UNA BASE ALL'ALTRA"CHR$ (17)
40 INPUT"INTRODUCI LA BASE DI PARTENZA";
A:PRINTCHR$ (17);
45 AA=54+A:IFA<11THENAA=47+A
50 IFA<2ORA>35ORINT(A)<>ATHENPRINT"ATTEN
TO - PROVA DI NUOVO":GOTO40
60 INPUT"INTRODUCI IL NUMERO DA CONVERTI
RE";N$:PRINTCHR$ (17);
70 REM VERIFICA CHE N$ E' NELLA FORMA GI
USTA
80 IFN$=""THENPRINT,"NON E' UN NUMERO":G
OTO60
90 L=LEN(N$):I=0
100 I=I+1:N=ASC (MID$ (N$,I,1))
110 IFN<48ORN>57ANDN<65)ORN>AATHENPRINT
,"NON E' UN NUMERO":GOTO60
```

```
120 IFI<LTHEN100
122 REM MEMORIZZA N$ IN UNA TABELLA
124 DIMA(L)
125 FORI=1TOL:N=ASC (MID$ (N$,I,1)):IFN<
58THENA(I)=N-48
126 IFN>64THENA(I)=N-55
127 NEXT
150 REM CONVERTE NUMERO DALLA BASE A ALL
A BASE 10
160 N=VAL(N$)
170 IFA<>10THENN=0:FORI=1TOL:N=A(I)+N*A:
NEXT
180 PRINT"FORMA DECIMALE DEL NUMERO"NCHR
$ (17)
190 INPUT"INTRODUCI LA BASE A CUI CONVER
TIRE";B:PRINTCHR$ (17)
200 IFB<2ORB>35ORINT(B)<>BTHENPRINT"ATTE
NTO - PROVA DI NUOVO":GOTO190
210 REM CONVERTI N IN BASE B
220 N$=""
230 M=INT(N/B):R=N-B*M:N=M
240 IFR<10THENN$=CHR$ (48+R)+N$
250 IFR>9THENN$=CHR$ (55+R)+N$
260 IFN<>0THEN230
270 PRINT"IL NUMERO IN BASE"B"E' "N$CHR$
 (17)
280 PRINT,"   ANCORA? S/N"
290 GETG$:IFG$<>"S"ANDG$<>"N"THEN290
300 IFG$="S"THENRUN
310 PRINTCHR$ (147),"   ARRIVEDERCI":END
```

I numeri del 64

Un numero tra 0 e 255 può essere rappresentato in binario usando un massimo di 8 coefficienti. Ad esempio

$$255 = 1*2^7 + 1*2^6 + 1*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 1*2 + 1$$
$$132 = 1*2^7 + 0*2^6 + 0*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 0*2 + 0$$

Il gruppo di questi 8 coefficienti, o bit, nel calcolatore è detto byte. Il Commodore 64 memorizza i numeri decimali usando 2 bytes chiamati byte alto e basso (high e low byte) indicando quello alto i multipli di 256. Ad esempio il

numero 999 può essere memorizzato con il byte alto 3 e il byte basso 231 perché 999 = 3*256+231. Memorizzare i numeri mediante bytes è come memorizzarli in base 256.

Con l'istruzione PEEK potete vedere come il 64 memorizza il byte alto e basso di un numero. Ricordatevi che le variabili intere su Commodore 64 sono specificate con il segno di percento (%) dopo il nome della variabile. Scrivete le righe che seguono, premendo il tasto <RETURN> alla fine di ogni linea.

```
NEW : CLR
X% = 999 (scrivete qui un numero intero)
PRINT"BYTE ALTO =" PEEK(2053), "BYTE BASSO =" PEEK(2054)
```

Il più grande intero che il Commodore 64 può memorizzare è 32767 che equivale a 127*256+255. I numeri con il byte alto uguale o maggiore di 128 sono negativi. In effetti i numeri negativi sono memorizzati considerando prima il valore ABSsoluto del numero, poi calcolando il byte alto e basso e sottraendo rispettivamente i valori a 255 e a 256. Perciò ad esempio, −1 avrà il byte alto a 255 e quello basso a 255.

Quanto segue vi mostra come calcolare un numero partendo dal byte alto e basso. Poniamo che H sia il byte alto e L il byte basso.

```
IF H<128 THEN NUMERO = H*256 + L
IF H>=128 THEN NUMERO = −(255−H)*256 + (256−L)
```

L'ultima linea equivale alla seguente.

```
IF H>=128 THEN NUMERO = H*256 + L − 256*256
```

Provate ad inserire i numeri con POKE nelle locazioni 2053 e 2054 e poi chiedete al 64 di dare PRINTX% confrontando la risposta. Ad esempio provate a scrivere le righe seguenti, premendo il tasto <RETURN> alla fine di ognuna.

```
NEW:CLR:X%=0
POKE2053,98:POKE2054,99 (inserite 98 e 99)
N=PEEK(2053)*256 + PEEK(2054)
IF PEEK(2053) >= 128 THEN N=N−256*256
PRINT X%,N
```

Numeri piccoli

Il programma dato nel primo paragrafo lavora solo con numeri interi positivi, ma un qualsiasi numero, intero o frazionario, ha una rappresentazione in qualsiasi base. Ad esempio il numero decimale 0.25 espresso in binario assume la forma 0.01, mentre il decimale 0.125 diventa 0.001 in binario. Per vedere questo, intendiamoci prima sul significato del numero decimale 0.25. Esso rappresenta due decimi e cinque centesimi, cioè

$$\begin{aligned} 0.25 &= 2/10 + 5/100 \\ &= 2*10^{-1} + 5*10^{-2} \end{aligned}$$

Esprimerlo in base B significa scriverlo nella forma

$$Y_1*B^{-1} + Y_2*B^{-2} + \ldots\ldots$$

dove come al solito $B^{-1} = 1/B$, $B^{-2} = 1/B^2$, etc.

I numeri decimali 0.25 e 0.125 possono essere scritti in questo modo:

$$\begin{aligned} 0.25 &= 1/4 \\ &= 0*2^{-1} + 1*2^{-2} \\ 0.125 &= 1/8 \\ &= 0*2^{-1} + 0*2^{-2} + 1*2^{-3} \end{aligned}$$

che spiega l'espressione binaria di questi numeri.

Come altro esempio, vediamo il numero 0.6 espresso in termini di potenze (negative) di 2.

$$\begin{aligned} 0.6 = {} & 1*2^{-1} + 0*2^{-2} + 0*2^{-3} + 1*2^{-4} \\ & + 1*2^{-5} + 0*2^{-6} + 0*2^{-7} + 1*2^{-8} \\ & + 1*2^{-9} + 0*2^{-10} + 0*2^{-11} + 1*2^{-12} \\ & + 1*2^{-13} + 0*2^{-14} + 0*2^{-15} + 1*2^{-16} \\ & + \ldots. \end{aligned}$$

In realtà, ci servono un numero infinito di termini per esprimere correttamente 0.6 in binario, assumendo la forma

0.1001 1001 1001 1001 1001 1001 1001 1001....

Il Commodore 64 memorizza solo 32 di questi numeri incominciando con il primo non nullo e arrotondando se il 33esimo non è nullo. Perciò il 64 memorizza 0.6 cosi

0.1001 1001 1001 1001 1001 1001 1001 1010

in forma binaria.

Il programma seguente visualizza la forma binaria dei numeri compresi tra 0 e 1 come è memorizzata dal Commodore 64.

```
10 REM DA DECIMALE A BINARIO
20 PRINTCHR$ (147),"DA DECIMALE A BINARI
O"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA STAMPA LA FORM
A BINARIA";
40 PRINT"DI UN NUMERO COMPRESO TRA 0 E 1
"CHR$ (17)
50 N=0:INPUT"SCRIVI IL TUO NUMERO";N:PRI
NT
60 IFN<=0ORN>=1THENPRINT"TRA 0 E 1 !!":G
OTO50
70 N$="0."
80 FORI=1TO32
90 N=N*2:N$=N$+MID$ (STR$(INT(N)),2,1):N
=N-INT(N)
100 NEXTI
110 PRINT"LA FORMA BINARIA DEL TUO NUMER
O E':"CHR$ (17)
120 PRINTN$CHR$ (17)
130 PRINT,"   ANCORA? S/N"
140 GETG$:IFG$<>"S"ANDG$<>"N"THEN140
150 IFG$="S"THENPRINT:GOTO50
160 PRINTCHR$ (147),"   ARRIVEDERCI":END
```

Virgola mobile

Gli interi nel Commodore 64 sono memorizzati usando 2 bytes, ma i numeri sono memorizzati con 5 bytes, anche se esso è un intero. Se non dichiarate intero un numero usando il segno %, esso verrà memorizzato come un numero reale usando 5 bytes. Un numero può essere espresso in forma binaria nel seguente modo:

$$1.X_1X_2X_3....X_M * 2^N$$

dove $X_1,X_2,....X_M$ sono 0 o 1 e N è un intero (positivo, negativo o nullo). L'intero N è detto l'*esponente binario* del numero e l'altra parte è detta *mantissa binaria.* Per esempio il numero decimale 10 in binario diventa 1010 che può essere riscritto

$$1.01 * 2^3$$

cosicché 10 ha esponente binario 3 e mantissa 1.01. Come altro esempio guardiamo il numero decimale 0.375 che è in binario 0.011 e può essere scritto

$$1.1 * 2^{-2}$$

cosicché il decimale 0.375 ha esponente binario –2 e mantissa binaria 1.1.

Abbiamo detto che il 64 usa 5 bytes per memorizzare il numero: il primo byte è l'esponente binario più 129 ed i rimanenti danno la mantissa binaria e il segno. Poiché il primo termine nella mantissa binaria è sempre 1, non dobbiamo memorizzarlo – semplicemente memorizziamo tutte le cifre a destra del punto decimale (secondo la notazione anglo-sassone) della mantissa binaria. Il primo bit del byte 2 contiene il segno del numero ed i rimanenti 31 bits degli altri 4 bytes (ignorando il primo bit) contengono la mantissa binaria.

Per esempio, il decimale 10 verrà memorizzato come segue: il primo byte è 129 più l'esponente binario 3, che fa 132. Il primo bit del secondo byte sarà 0 perché il numero è positivo ed i restanti 31 bit saranno

01000000000000000000000000000000

Poichè la mantissa binaria è 1.01 e noi ignoriamo l'1 che precede il punto decimale. Perciò i 32 bit degli ultimi 4 bytes saranno

00100000000000000000000000000000

che, divisi in gruppi di 8, danno

00100000 00000000 00000000 00000000

che nell'ordine valgono 32,0,0,0. Perciò i 5 bytes usati per memorizzare il decimale 10 saranno 132,32,0,0,0.

Possiamo invertire il procedimento e trovare quale numero il 64 contiene nei 5 bytes. Supponiamo che il numero N sia memorizzato nei cinque bytes P,Q,R,S,T: le seguenti linee di programma calcolano N a partire da P,Q,R,S e T.

```
X = 1 : IF Q>= 128 THEN Q=Q-128:X=-1
N = X*2^(P-129)*(1+Q*2^-7+R*2^-15+S*2^-23+T*2^-31)
```

Per vedere il 64 all'opera, scrivete le seguenti linee, premendo il tasto RETURN> dopo ognuna.

```
NEW : CLR
X=10 (scrivete qui un qualsiasi numero)
FOR I=0 TO 4 : PRINT "BYTE" I+1 "=" PEEK(2053 +I)
NEXT I
```

In realtà, se provate questo programmino, scoprirete che il Commodore compie alcuni piccoli errori quando esegue questa moltiplicazione. Il numero $1+2^{-24}$ è memorizzato (correttamente) con i seguenti 5 bytes:

129,0,0,0,128

Comunque lo stesso numero scritto nella forma $1*(1+2^{-24})$ è memorizzato con i seguenti 5 bytes:

129,0,0,0,64

In altre parole, stando al Commodore 64

$1*(1 + 2\uparrow(-24)) = 1 + 2\uparrow(-25)$

Un altro modo di visualizzare lo stesso problema è il seguente:

```
X = 1+2↑-24
PRINT X - X,X - 1*X
```

In alternativa, potete provare con:

```
X = 1+2↑-24
X1 = 1*X
X2 = 1*X1
X3 = 1*X2
X4 = 1*X3
PRINTX,X1,X2,X3,X4
```

Problemi di questa natura sembra che non compaiano se i 3 byte centrali, dei 5 usati per memorizzare il numero, non sono tutti uguali a 0. Cercate voi qualche esempio.

CAPITOLO 7

GIORNI E SETTIMANE

Giorni

La congruenza di Zeller è una complicata formula di previsione che calcola i giorni della settimana (Lunedì, Martedì, etc.) per una qualsiasi data. Con questa formula potreste, per esempio, trovare in quale giorno della settimana una persona è nata ed anche, se la vostra memoria non è buona, sapere in quale giorno della settimana cade un anniversario.

Questa che segue è la formula di Zeller:

```
Z = INT(2.6*M-0.1) + G + A + INT(A/4) + INT(S/4) - 2*S
X = Z - 7*INT(Z/7)
```

Il numero X è compreso tra 0 e 6 perché tutti i multipli di 7 più piccoli di A sono sottratti da A. Questi numeri rappresentano i giorni della settimana nel seguente modo:

0 : Domenica
1 : Lunedì
2 : Martedì
3 : Mercoledì
4 : Giovedì
5 : Venerdì
6 : Sabato

I numeri G, M, A e S sono definiti come segue:

G: giorno del mese;

M: il numero del mese - ma non il numero corrente perché Gennaio e Febbraio sono i numeri 11 e 12 dell'anno precedente (con conseguenze sui valori di A e S descritti più avanti). Marzo è il numero 1, Aprile il 2, Maggio il 3 e così via fino a Dicembre il 10;

A: l'anno del secolo;

S: il numero dei secoli, ovvero le prime due cifre dell'anno.

Ad esempio, se la data è il 26 Agosto 1983 si scrive 26/08/83 ma per la formula di Zeller useremo G=26, M=6, A=83, e S=19.

Sostituendo tali valori nella formula, si ottiene:

```
Z = INT(2.6*6-0.1)+26+83+INT(83/4)+INT(19/4)-2*19
  = INT(15.6-0.1)+26+83+INT(20.75)+INT(4.75)-38
  = 15+26+83+20+4-38
  = 110
```

e

```
X = 110-7*INT(110/7)
  = 110-7*INT(15.714285)
  = 110-105
  = 5.
```

Perciò concludiamo che il 26 Agosto 1983 era Venerdì.

Ecco qui di seguito altri esempi che mostrano la notazione normale della data e quella per la formula di Zeller.

NOTAZIONE NORMALE	NOTAZIONE DI ZELLER			
	G	M	S	A
03/03/1947	3	1	19	47
01/01/2000	1	11	19	99
26/02/1983	26	12	19	82
29/11/1984	29	9	19	84

Il programma che segue usa la formula di Zeller per il calcolo del giorno della settimana per una qualsiasi data. Il vostro Commodore 64 calcola i giusti valori di G,M,S,A richiesti per la data che voi introdurrete. Si osservi che il programma esegue alcuni controlli perché la data da voi introdotta abbia senso e perciò il 30 Febbraio 1983 non sarà accettata. Inoltre l'anno deve essere un intero compreso tra 1582 e 4902. Infatti il Calendario Giuliano fu introdotto nel 1582 in Europa, ma in Inghilterra, nelle Colonie Inglesi e negli Stati Uniti fu introdotto solo nel 1752.

Il programma tiene conto automaticamente degli anni bisestili considerando che è bisestile l'anno il cui numero è divisibile esattamente per 4, o per 400 purché non sia divisibile anche per 100. Perciò il 1900 non fu un anno bisestile, mentre lo sarà il 2000.

```
10 REM GIORNO DELLA SETTIMANA
20 DIMA(12),A$(12):FORI=1TO12:READA(I),A
$(I):NEXTI
30 FORI=0TO6:READB$(I):NEXTI
40 PRINTCHR$ (147)"          GIORNO DELLA
SETTIMANA"CHR$ (17)
50 PRINT"QUESTO PROGRAMMA CALCOLA IL GIO
RNO DELLA"
60 PRINT"SETTIMANA PER UNA QUALSIASI DAT
A SPECI- FICATA"CHR$ (17)
70 PRINT"SCRIVI LA DATA:"CHR$ (17)
80 A(2)=29:D=0:INPUT"GIORNO, DA 1 A 31:
";D:PRINT
90 IFD<1ORD>31ORD<>INT(D)THENPRINT,"PROV
A ANCORA":GOTO80
100 M=0:INPUT"MESE, DA 1 A 12 : ";M:PRIN
T
110 IFM<1ORM>12ORM<>INT(M)THENPRINT,"PRO
VA ANCORA":GOTO100
120 IFD>A(M)THENPRINT"IL MESE NON HA"D"G
IORNI:PROVA ANCORA":GOTO80
130 Y=0:INPUT"ANNO : ";Y:PRINT
140 IFY<1582ORY>4902THENPRINT,"NON E' NE
L PERIODO AMMESSO":GOTO130
150 IFY<>INT(Y)THENPRINT,"NON E' UN ANNO
, PROVA ANCORA":GOTO130
160 REM VERIFICA ANNI BISESTILI
170 L=0:IFINT(Y/4)*4=YTHENL=-1
180 IFLANDINT(Y/100)*100=YTHENL=0:IFINT(
Y/400)*400=YTHENL=-1
190 A(2)=28-L:IFLTHENPRINT"QUESTO E' UN
ANNO BISESTILE"CHR$ (17)
200 IFM=2ANDD>A(2)THENPRINT"28 GIORNI IN
 FEBBRAIO":GOTO80
210 PRINT"IL";D;A$(M);Y;"E' / ERA ";
220 REM CAMBIA FORMATO DEI DATI INTRODOT
TI
230 M=M-2:IFM<1THENM=M+12:Y=Y-1
240 C=INT(Y/100):Y=Y-C*100
250 REM CONGRUENZA DI ZELLER
260 DAY=INT(2.6*M-0.1)+D+Y+INT(C/4)+INT(
Y/4)-2*C
```

```
270 DAY=DAY-7*INT(DAY/7):PRINTB$(DAY)CHR
$ (17)
280 PRINT,"    ANCORA? S/N"
290 GETG$:IFG$<>"S"ANDG$<>"N"THEN290
300 IFG$="S"THENPRINT:GOTO70
310 PRINTCHR$ (147),"    ARRIVEDERCI":END
400 REM DATA
410 DATA31,GENNAIO,29,FEBBRAIO,31,MARZO,
30,APRILE
420 DATA31,MAGGIO,30,GIUGNO,31,LUGLIO,31
,AGOSTO
430 DATA30,SETTEMBRE,31,OTTOBRE,30,NOVEM
BRE,31,DICEMBRE
440 DATADOMENICA,LUNEDI',MARTEDI',MERCOL
EDI',GIOVEDI',VENERDI',SABATO
```

Nota: È possibile che voi abbiate già visto la formula di Zeller scritta in modo leggermente diverso. Spesso il primo termine è scritto INT(2.6*M–0.2) anziché INT(2.6*M–0.1) e questa seconda forma è usata a causa del modo in cui il Commodore 64 calcola la parte INTera del numero. Provate con M=7 ed avrete

$$\begin{aligned} \text{INT}(2.6*7-0.2) &= \text{INT}(18.2 - 0.2) \\ &= \text{INT}(18) \\ &= 18 \end{aligned}$$

Comunque il Commodore 64 vi dà 17 come valore di INT(2.6*7–0.2) anche se chiedendogli 2.6*7–0.2 vi stampa 18 correttamente.

Calendario

Una volta che noi conosciamo il giorno della settimana, possiamo produrre un calendario, come fa il programma che segue per qualsiasi mese di qualsiasi anno. (Solo un mese per volta può essere mostrato chiaramente sullo schermo). Il programma calcola con la formula di Zeller il giorno della settimana in cui cade il primo giorno del mese. Il resto dei giorni prosegue automaticamente. Come nel programma Giorni della Settimana, gli anni bisestili sono conteggiati automaticamente.

```
10 REM CALENDARIO
20 DIMA(12),A$(12):FORI=1TO12:READA(I),A
$(I):NEXTI
30 REM L'INIZIO
40 PRINTCHR$ (147),"      CALENDARIO"CHR$
 (17)
50 PRINT"QUESTO PROGRAMMA STAMPA UN CALE
NDARIO"
60 PRINT"PER UN QUALSIASI MESE DI UN QUL
SIASI ANNO"CHR$ (17)
70 PRINT"SCRIVI IL MESE E L'ANNO :"CHR$
(17)
80 M=0:INPUT"MESE, DA 1 A 12: ";M:PRINT
90 IFM<1ORM>12ORM<>INT(M)THENPRINT,"PROV
A DI NUOVO":GOTO80
100 Y=0:INPUT"ANNO :";Y:PRINT
110 IFY<1582ORY>4902THENPRINT,"NON E' NE
L PERIODO AMMESSO":GOTO100
120 IFY<>INT(Y)THENPRINT,"NON E' UN ANNO
, PROVA ANCORA":GOTO100
130 REM VERIFICA ANNO BISESTILE
140 L=0:IFINT(Y/4)*4=YTHENL=-1
150 IFLANDINT(Y/100)*100=YTHENL=0:IFINT(
Y/400)*400=YTHENL=-1
160 A(2)=28-L:IFLTHENPRINT"QUESTO E' UN
ANNO BISESTILE"CHR$ (17)
170 A$=A$(M)+STR$(Y):MM=A(M)
200 REM CAMBIA FORMATO DEI DATI DI INPUT
210 M=M-2:IFM<1THENM=M+12:Y=Y-1
220 C=INT(Y/100):Y=Y-C*100
230 REM CONGRUENZA DI ZELLER
240 DAY=INT(2.6*M-0.1)+1+Y+INT(C/4)+INT(
Y/4)-2*C
250 DAY=DAY-7*INT(DAY/7)
260 REM STAMPA CALENDARIO
270 PRINTCHR$ (147):PRINT:POKE53281,7
280 PRINTTAB(20-LEN(A$)/2)CHR$ (31)A$CHR
$ (17)
290 PRINTCHR$ (28)"   DOM  LUN  MAR  MER
  GIO  VEN  SAB"CHR$ (31)
300 FORI=1TOMM
310 DAY=DAY+1
```

```
320 PRINTTAB(DAY*5-2+(I>9));I;
330 IFDAY>6THENDAY=0:PRINT:PRINT
340 NEXTI
350 PRINT:PRINT:PRINT
360 PRINT,CHR$ (5)"     ANCORA? S/N"CHR$
(31)
370 GETG$:IFG$<>"S"ANDG$<>"N"THEN370
380 IFG$="S"THENPRINT:GOTO70
390 PRINTCHR$ (147),"     ARRIVEDERCI":EN
D
400 REM DATA
410 DATA31,GENNAIO,29,FEBBRAIO,31,MARZO,
30,APRILE
420 DATA31,MAGGIO,30,GIUGNO,31,LUGLIO,31
,AGOSTO
430 DATA30,SETTEMBRE,31,OTTOBRE,30,NOVEM
BRE,31,DICEMBRE
```

Gestione delle date

Talvolta può essere necessario conoscere un elenco di date che sono lontane un certo numero di giorni da un'altra: ad esempio i giorni per certi trattamenti ospedalieri o i giorni di scadenza di pagamenti.

Per produrre un tale elenco useremo la data 'Pseudo-Giuliana' che è semplicemente il numero di giorni a partire da una data nota. (In pratica il primo di Gennaio dell'anno ha una data pseudo-Giuliana di 1). Una formula molto semplice esegue la conversione tra i due tipi di date.

Se la data è G/M/A, essendo G il giorno, M il mese e A l'anno, allora la data pseudo-Giuliana è calcolata così:

X=INT(30.57*M)+INT(365.25*A−395.25)+G
Se M>2 ed A è bisestile si sottrae 1 da X
Se M>2 ed A non è bisestile si sottrae 2 da X

Come esempio calcoliamo la data pseudo-Giuliana del 26 Agosto 1983. I valori di G, M ed A sono dati da G=26, M=8 ed A=1983 che, sostituiti nella formula, danno:

X = INT(30.57*8) + INT(365.25*1983−395.25)+26
= INT(244.56) + INT(723895.5)+26
= 244+723895+26
= 724165

Alla fine, poiché il mese M è maggiore di 2 e l'anno A non è bisestile, sottraiamo 2 da X e otteniamo la data pseudo-Giuliana 724163.

Invece, per calcolare la data partendo da quella pseudo-Giuliana, si fa così:

La prima approssimazione dell'anno è
A = INT(X/365.26) + 1

Il giorno dell'anno è
G = X − INT(365.25*A−395.25)

La correzione per l'anno bisestile è
G1 = 2, se l'anno è bisestile G1=1
Se G>91−G1 aggiungere G1 a G

Il calcolo del giorno e del mese è
M = INT(G/30.57)
G = G − INT(30.57*M)

La correzione del mese e dell'anno è
Se M>12, porre M=1 e aggiungere 1 ad A

Come esempio calcoliamo la data corrispondente alla data pseudo-Giuliana del 724164 (che è 1 più di quella calcolata prima). I calcoli si sviluppano così:

A = INT(724164/365.26) + 1
= INT(1982.5987) + 1
= 1983

G = 724164 − INT(365.25*1983−395.25)
= 724164 − INT(724290.75−395.25)
= 724164 − INT(723895.5)
= 269

Il 1983 non è bisestile perciò G1=2. Il valore di G è maggiore di 91−G1 e perciò aggiungiamo G1 a G. Il valore di G è ora dato da

G = 271

M = INT(271/30.57)
= INT(8.86490023)
= 8

$$G = 271 - INT(30.57*8)$$
$$= 271 - INT(244.56)$$
$$= 271 - 244$$
$$= 27$$

Poiché il valore di M non è maggiore di 12, abbiamo finito con G=27, M=8 e A=1983. Quindi il 27 Agosto 1983 corrisponde alla data pseudo-Giuliana 724164.

Il programma successivo esegue tutti questi calcoli rapidamente e fornisce un elenco di date che sono lontane un certo numero di giorni tra loro. Per semplicità, il programma lavora solo con date del ventesimo secolo, ma voi sarete certo in grado di apportare tutte le modifiche per un altro secolo.

Attenzione. Le date sono introdotte nella forma GG/MM/AA, ad esempio il 12 Marzo 1984 deve essere introdotto 12/03/84 o 12/3/84 ma non 12/ 3/84. Non vengono eseguiti molti controlli sulla congruenza della data e perciò è possibile introdurre il 30/02/84. Il programma considererà quella data come il 2 Marzo 1983 (riuscite a capirne il perché?).

```
10 REM GESTIONE DELLE DATE
20 POKE53281,7:PRINTCHR$ (147),CHR$ (28)
"  GESTIONE DELLE DATE"CHR$ (17)
30 PRINTCHR$ (31)"QUESTO PROGRAMMA PROVV
EDE AD ELENCARE"
40 PRINT"DATE CHE DISTANO UN NUMERO SPEC
IFICO DI ";
50 PRINT"GIORNI"CHR$ (17)
60 REM INTRODUZIONE DATA DI PARTENZA
70 PRINT"SCRIVI LA DATA COME GG/MM/AA":P
RINT"< ESEMPIO 15/02/84 >"
80 S$="":INPUT"INTRODUCI DATA DI PARTENZ
A: ";S$:PRINT
90 IFLEN(S$)<>8THENPRINT,"NELLA FORMA GG
/MM/AA":GOTO80
100 IFMID$ (S$,3,1)<>"/"ORMID$ (S$,6,1)<
>"/"THENPRINT,"USA GG/MM/AA":GOTO80
110 D=VAL(MID$ (S$,1,2)):M=VAL(MID$ (S$,
4,2)):Y=VAL(MID$ (S$,7,2))+1900
120 IFD<=0ORD>31ORM<=0ORM>12THENPRINT,"E
RRORE DI GIORNO E/0 MESE":GOTO80
```

```
130 REM LA DATA E' ORA IN FORMA CORRETTA
140 REM CALCOLA LA DATA PSEUDO-GIULIANA
150 X=INT(30.57*M)+INT(365.25*Y-395.25)+
D
160 REM CORREZIONE PER L'ANNO BISESTILE
170 IFM>2THENX=X-2:IFINT(Y/4)*4=YTHENX=X
+1
180 REM INTRODUZIONE INTERVALLO DI GIORN
I
190 PRINT"INTRODUCI L'INTERVALLO DI GIOR
NI TRA LE DATE"
200 P=0:INPUT"INTERVALLO : ";P:PRINT
210 IFP<=0ORINT(P)<>PTHENPRINT,"SCRIVI U
N NUMERO INTERO":GOTO200
220 PRINT"NUMERO DI INTERVALLI DI TEMPO
RICHIESTI"
230 N=0:INPUT"NUMERO : ";N:PRINT
240 IF INT(N)<>NTHENPRINT,"UN NUMERO INT
ERO":GOTO230
250 IFN<=0ORN>100THENPRINT,"SII RAGIONEV
OLE":GOTO230
260 PRINTCHR$ (28):REM COLORE PER STAMPA
RE LE DATE
270 FORI=0TON
280 :Y=INT(X/365.26)+1:REM ANNO
290 :D=X-INT(365.25*Y-395.25)
300 :REM CORREGGI ANNO BISESTILE
310 :D1=2:IFINT(Y/4)*4=YTHEND1=1
320 :IFD>91-D1THEND=D+D1
330 :M=INT(D/30.57):REM MESE
340 :D=D-INT(30.57*M):REM GIORNO
350 :IFM>12THENM=1:Y=Y+1
360 :Y=Y-1900
370 :REM VISUALIZZAZIONE
380 :Z=D:GOSUB500:D$=Z$+"/"
390 :Z=M:GOSUB500:D$=D$+Z$+"/"
400 :Z=Y:GOSUB500:D$=D$+Z$
410 :PRINTD$,;
420 :X=X+P:REM AGGIUNGE L'INTERVALLO
430 NEXT
440 PRINT:PRINT:PRINT,CHR$ (5)"   ANCORA
? S/N"CHR$ (31)
```

```
450 GETG$:IFG$<>"S"ANDG$<>"N"THEN450
460 IFG$="S"THENPRINT:RUN
470 PRINTCHR$ (147),"     ARRIVEDRCI":END
500 REM SOTTOPROGRAMMA DI FORMATO
510 Z$=MID$ (STR$(Z),2):IFLEN(Z$)<2THENZ
$="0"+Z$
520 RETURN
```

CAPITOLO 8

MASSIMO COMUN DIVISORE (MCD)

Se A e B sono numeri interi, allora un *divisore comune* (o *fattore comune)* di A e B è un intero che divide entrambi i numeri e il *massimo* comun divisore (o *massimo* fattore comune) di A e B è il più grande di tali interi.

Ad esempio 3 è un divisore comune di 12 e 18, ma 6 è il massimo comun divisore (MCD) di 12 e 18.

Il calcolo del massimo comun divisore di un numero non è molto complicato, soprattutto per un calcolatore ed il metodo seguito serve come bell'esempio di algoritmo di calcolo.

L'*algoritmo di Euclide* è il più noto e antico (terzo secolo a.C.) metodo di calcolo del MCD. La procedura di ricerca è questa:

1. Riassegnate (se necessario) A e B in modo che A sia maggiore di B
2. Dividete A per B e trovate il resto R_1

   $R_1 = A - B*INT(A/B)$

Notate che ogni numero, che divide A e B, divide anche R_1 e che per conseguenza ogni divisore comune di B e R_1 è anche un divisore di A. Ne segue che i divisori comuni di A e B solo gli stessi di B e R_1 e perciò il MCD di A e B è uguale al MCD di B e R_1.

3. Ora dividete B per R_1 e trovate il resto R_2.
   $R_2 = B - R_1 * INT(B/R_1)$

Le considerazioni fatte sopra per B e R_1 si applicano anche a R_1 e R_2 e perciò il MCD di A e B è anche il MCD di R_1 e R_2.

4. Ancora, dividere R_1 per R_2 per ottenere il resto R_3.
   $R_3 = R_1 - R_2 * INT(R_1/R_2)$

Il procedimento prosegue finché il resto è zero. Notate che i resti decrescono ad ogni calcolo e quindi lo zero è raggiunto dopo un certo numero di passi.

$R_1 = A - B*INT(A/B)$ $(0<=R_1<B)$

$R_2 = B - R_1*INT(B/R_1)$ $(0<=R_2<R_1)$

$R_3 = R_1 - R_2*INT(R_1/R_2)$ $(0<=R_3<R_2)$

.....

.....

.....

$R_{N-1} = R_{N-3} - R_{N-2}*INT(R_{N-3}/R_{N-2})$ $(0<=R_{N-1}<R_{N-2})$

$R_N = R_{N-2} - R_{N-1}*INT(R_{N-2}/R_{N-1})$ $(R_N=0)$

Quando il resto è zero, vediamo che il precedente resto R_{N-1} è il MCD di R_{N-1} e R_{N-2} e, ragionando in questo modo, otteniamo che R_{N-1} è il MCD di A e B.

Il procedimento evidenziato sopra è facilmente impostato sul calcolatore, come si vede nel programma seguente.

```
10 REM MASSIMO COMUN DIVISORE
20 PRINTCHR$ (147)"          MASSIMO COMUN
 DIVISORE"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA CALCOLA IL MCD
 DI DUE"
40 PRINT"INTERI USANDO L'ALGORITMO DI EU
CLIDE"CHR$ (17)
50 PRINT"INTRODUCI I DUE INTERI"CHR$ (17
)
60 INPUT"  PRIMO INTERO ";A:PRINT
70 IFA<1ORINT(A)<>ATHENPRINT"PROVA ANCOR
A":GOTO60
80 INPUT"SECONDO INTERO ";B:PRINT
90 IFB<1ORINT(B)<>BTHENPRINT"PROVA ANCOR
A":GOTO80
100 IFA<BTHENC=A:A=B:B=C
110 REM ALGORITMO DI EUCLIDE
120 R=A:S=B
130 T=R-S*INT(R/S)
140 IFT<>0THENR=S:S=T:GOTO130
150 PRINT"IL MASSIMO COMUN DIVISORE E':"
:PRINT,,CHR$ (158);S
160 PRINTCHR$ (154)"IL MINIMO COMUNE MUL
TIPLO E':":PRINT,,CHR$ (158);A*B/S
```

```
170 PRINT:PRINTCHR$ (154),"    ANCORA? S/
N"
180 GETG$:IFG$<>"S"ANDG$<>"N"THEN180
190 IFG$="S"THENRUN
200 PRINTCHR$ (147),"    ARRIVEDERCI":END
```

Il programma calcola anche il minimo comune multiplo (MCM) di A e B. Il minimo comune multiplo di due numeri A e B è il più piccolo numero che è divisibile per A e B contemporaneamente ed è dato da

A * B / (MCD)

Se D è il MCD di A e B, allora è possibile scrivere D come combinazione di A e B:

D = S*A + T*B

dove S e T sono interi e possono essere calcolati utilizzando al contrario l'algoritmo di Euclide. Provate a modificare il programma MCD per calcolare anche T e S.

CAPITOLO 9
NUMERI PRIMI

Un numero *primo* è un intero (maggiore di 1) che non è divisibile per alcun altro numero escluso 1 e se stesso (è chiaro che per divisibilità intendiamo divisibilità esatta). I numeri 2, 3, 5, 7 e 11 sono primi mentre 4, 6, 8, 9 e 10 non lo sono. I numeri non primi sono detti *composti.*

Non siste una formula generale per ottenere i numeri primi, ma Euclide dimostrò (nel 300 a.C.) che esistono infiniti numeri primi e noi sappiamo anche che ve ne sono sempre meno fra i numeri grandi.

La ricerca dei numeri primi è di notevole interesse e soprattutto lo è diventata in tempi recenti per motivi di crittografia.

Un semplice metodo diretto, per determinare se un numero N è primo, è il cosiddetto Crivello di Eratostene (Eratostene di Cirene fu un matematico greco vissuto tra il 276 e il 196 a.C., che calcolò anche la circonferenza della Terra). L'idea è quella di scrivere tutti i numeri da 1 a N. Poi si lascia il 2 e si cancellano tutti i numeri pari seguenti. Il primo numero non cancellato dopo il 2 è primo ed è il 3. Adesso cancelliamo tutti i multipli di 3 ed il numero successivo a 3 non cancellato è il 5 che deve essere primo. Ora cancelliamo tutti i multipli di 5 e continuiamo così: quelli che rimangono sono i numeri primi tra 1 e N.

La tabella qui sotto mostra il risultato con un crivello sui numeri fino a 100. I multipli di 2 sono segnati con /, quelli di 3 con —, quelli di 5 con \, e quelli di 7 con |.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Il programma che segue usa il Crivello di Eratostene per produrre l'elenco dei numeri primi minori di un certo numero N che è chiesto all'inizio del programma.

```
10 REM CRIVELLO DI ERATOSTENE
20 PRINTCHR$ (147)"          CRIVELLO DI E
RATOSTENE"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA PUO' CALCOLARE
 I NUMERI";
40 PRINT"PRIMI FINO AD UN CERTO NUMERO D
ATO"CHR$ (17)
50 INPUT"FINO A QUALE NUMERO ";N:PRINT
60 IFN<4ORINT(N)<>NORN>15000THENPRINT"SI
I RAGIONEVOLE":GOTO50
70 DIMA%(N):PRINTCHR$ (158)"NUMERI PRIMI
 DA 2 A";N:PRINT
80 FORI=2TON
90 IFA%(I)=1THEN130
100 IF 39-POS(0)<LEN(STR$(I))THENPRINT
110 PRINTI;
120 FORJ=ITONSTEPI:A%(J)=1:NEXT
130 NEXT:PRINT
140 PRINT:PRINTCHR$ (154),"    ANCORA? S/
N"
150 GETG$:IFG$<>"S"ANDG$<>"N"THEN150
160 IFG$="S"THENRUN
170 PRINTCHR$ (147),"    ARRIVEDERCI":END
```

Si noti che il programma ha un avvio lento, ma ben presto inizia a scrivere velocemente i numeri primi. Impiega circa 7 minuti e 21 secondi per stampare i numeri primi tra 2 e 15000.

Il Crivello di Eratostene è concettualmente semplice ed è utile se volete un elenco di numeri primi, ma non è molto pratico come metodo per controllare se un certo numero è primo.

Un semplice metodo di controllo consiste nel verificare passo-passo se il numero è divisibile per numeri minori di N, come è illustrato nel programma seguente. Un cronometro, inserito nel programma, indica il tempo necessario per il controllo di primalità.

```
10 REM NUMERI PRIMI VERSIONE 1
20 PRINTCHR$(147)"PROVA PRIMI INEFFICIE
NTE - VERSIONE 1"CHR$(17)
30 INPUT"NUMERO DA PROVARE";N:PRINT
40 IFN<4ORN<>INT(N)THENPRINT"SII RAGIONE
VOLE":GOTO30
50 A$="":TI$="000000":X=N-1
60 FORI=2TOX
70 T=N/I:IFINT(T)=TTHENA$="NON ":I=X
80 NEXT
90 PRINT"IL NUMERO "A$"E' PRIMO"
100 PRINT:PRINT"TEMPO NECESSARIO PER IL
TEST DEL NUMERO";
105 PRINTINT(TI/60+0.5)"SECONDI"
110 PRINT:PRINTCHR$(154),"   ANCORA? S/
N"
120 GETG$:IFG$<>"S"ANDG$<>"N"THEN120
130 IFG$="S"THENRUN
140 PRINTCHR$(147),"   ARRIVEDERCI":END
```

Il programma funziona ma è veramente inefficiente e lento perchè, ad esempio, il controllo su 9001 richiede 113 secondi. Un po' di ragionamento può apportare notevoli vantaggi.

Per incominciare, non ci dobbiamo annoiare ad usare tutti i numeri tra 2 e N, ma dobbiamo considerare solo quelli compresi tra 2 e INT(SQR(N)). Se M è un intero che divide N ed è maggiore di INT(SQR(N)) allora N/M è un intero che divide N ed è minore di INT(SQR(N)). Questa semplice aggiunta è inserita nel programma che segue.

```
10 REM NUMERI PRIMI VERSIONE 2
20 PRINTCHR$(147)"PROVA PRIMI INEFFICIE
NTE - VERSIONE 2"CHR$(17)
30 INPUT"NUMERO DA PROVARE ";N:PRINT
40 IFN<4ORN<>INT(N)THENPRINT"SII RAGIONE
VOLE":GOTO30
50 A$="":TI$="000000":X=SQR(N)
60 FORI=2TOX
70 T=N/I:IFINT(T)>=TTHENA$="NON ":I=X
80 NEXT
90 PRINT"IL NUMERO "A$"E' PRIMO"
```

```
100 PRINT:PRINT"TEMPO NECESSARIO PER LA
VERIFICA:"
105 PRINTINT((TI*10)/60+0.5)/10"SECONDI"
110 PRINT:PRINTCHR$ (154),"   ANCORA? S/
N"
120 GETG$:IFG$<>"S"ANDG$<>"N"THEN120
130 IFG$="S"THENRUN
140 PRINTCHR$ (147),"   ARRIVEDERCI":END
```

Questa versione lavora molto più velocemente. La verifica di 9001 richiede ora solo 1 secondo mentre per un numero più grande come 987654323 ci vogliono 387 secondi. Non tentate di controllare un numero così grande con la versione 1 – altrimenti vi divertirete a guardare stupidamente lo schermo televisivo.

Nota. L'aggiunta inserita nel programma Primi Versione 2 può essere introdotta anche nel programma Eratostene 2. Con questa aggiunta impiega solo 6 minuti e 21 secondi per stampare i numeri primi tra 2 e 15000.

```
10 REM CRIVELLO DI ERATOSTENE : CON AGGI
UNTE
20 PRINTCHR$ (147)"         CRIVELLO DI E
RATOSTENE"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA PUO' CALCOLARE
 I NUMERI";
40 PRINT"PRIMI FINO AD UN CERTO NUMERO D
ATO"CHR$ (17)
50 INPUT"FINO A QUALE NUMERO ";N:PRINT
60 IFN<4ORINT(N)<>NORN>15000THENPRINT"SI
I RAGIONEVOLE":GOTO50
70 DIMA%(N):PRINTCHR$ (158)"NUMERI PRIMI
 DA 2 A";N:PRINT:S=INT(SQR(N)+1)
80 FORI=2TON
90 IFA%(I)=1THEN130
100 IF39-POS(0)<LEN(STR$(I))THENPRINT
110 PRINTI;
120 IFI<=STHENFORJ=ITONSTEPI:A%(J)=1:NEX
T
130 NEXT:PRINT
140 PRINT:PRINTCHR$ (154),"   ANCORA? S/
N"
```

```
150 GETG$:IFG$<>"S"ANDG$<>"N"THEN150
160 IFG$="S"THENRUN
170 PRINTCHR$ (147),"    ARRIVEDERCI":END
```

Possiamo ancora migliorare il programma primi Versione 2 prendendo spunto dal Crivello di Eratostene. Possiamo tralasciare tutti i numeri pari maggiori di 2 e tutti i multipli di 3. Queste aggiunte sono inserite nel programma che segue.

```
10 REM NUMERI PRIMI VERSIONE 3
20 PRINTCHR$ (147),"PROVA PRIMI VERSIONE
 3"CHR$ (17)
30 INPUT"NUMERO DA VERIFICARE ";N:PRINT
40 IFN<40RN<>INT(N)THENPRINT"SII RAGIONE
VOLE":GOTO30
50 A$="":TI$="000000":X=SQR(N)
52 IFINT(N/2)=N/2THENA$="NON ":GOTO90
53 IFINT(N/3)=N/3THENA$="NON ":GOTO90
60 FORI=5TOXSTEP6
70 T=N/I:IFINT(T)=TTHENA$="NON ":I=X
75 T=N/(I+2):IFINT(T)=TTHENA$="NON ":I=X
80 NEXT
90 PRINT"IL NUMERO "A$"E' PRIMO"
100 PRINT:PRINT"TEMPO NECESSARIO PER LA
VERIFICA:"
105 PRINTINT(TI*10/60+0.5)/10"SECONDI"
110 PRINT:PRINTCHR$ (154),"    ANCORA? S/
N"
120 GETG$:IFG$<>"S"ANDG$<>"N"THEN120
130 IFG$="S"THENRUN
140 PRINTCHR$ (147),"    ARRIVEDERCI":END
```

Queste aggiunte aumentano la velocità di esecuzione: ora la verifica di 987654323 richiede circa 133 secondi cioè un tempo ragionevole.

Un numero composto può essere scritto come prodotto di numeri primi. Ad esempio

6 = 3*2
24 = 3*2*2*2
81018001 = 9001*9001

e così via. I vari numeri primi che intervengono sono detti *fattori* del numero. Aggiungendo poche righe in più alla versione 3, possiamo visualizzare tutti i fattori di un numero così come é fatto dal programma che segue.

```
10 REM FATTORI PRIMI
20 GOTO70:REM LA PERTE CHE SEGUE E' UNAS
OTTOPROGR. DI CALCOLO DEI FATTORI
30 Y=N/T:IF38-POS(0)<LEN(STR$(Y))THENPRI
NT
40 PRINTY;:N=T:T=N/Y:IFT=INT(T)THEN30
50 S=SQR(N)+1:RETURN
60 REM PARTE PRINCIPALE
70 PRINTCHR$ (147),CHR$ (154)"   FATTORI
 PRIMI"CHR$ (17)
80 INPUT"NUMERO DA VERIFICARE ";N:PRINT
90 IFN<4ORN<>INT(N)THENPRINT"SII RAGIONE
VOLE":GOTO80
100 PRINTCHR$ (5)"I FATTORI DI";N;"SONO"
CHR$ (17)
110 TI$="000000":X=SQR(N):S=X+1
120 T=N/2:IFINT(T)=TTHENGOSUB30
130 T=N/3:IFINT(T)=TTHENGOSUB30
140 FORI=5TOXSTEP6
150 T=N/I:IFINT(T)=TTHENGOSUB30
160 T=N/(I+2):IFINT(T)=TTHENGOSUB30
170 IFI>STHENI=X
180 NEXT
190 IF38-POS(0)<LEN(STR$(N))THENPRINT
200 IFN>1THENPRINTN;
210 PRINT:PRINT:PRINTCHR$ (158)"E' TUTTO
"
220 PRINT:PRINT"IL TEMPO PER TROVARE I F
ATTORI E' STATO:"
225 PRINTINT(TI/6+0.5)/10"SECONDI"
230 PRINT:PRINTCHR$ (154)."   ANCORA? S/
N"
240 GETG$:IFG$<>"S"ANDG$<>"N"THEN240
250 IFG$="S"THENRUN
260 PRINTCHR$ (147)."   ARRIVEDERCI":END
```

Se introducete un numero molto grande nel programma, il Commodore 64 lo arrotonda e trova i fattori di quest'ultimo numero.

Numeri Primi Grandi

La ricerca di numeri primi molto grandi è diventato per alcuni un passatempo. I numeri primi più grandi che si conoscono sono normalmente numeri di Mersenne. I numeri nella forma

$$2^p - 1$$

sono detti *numeri di Mersenne* dal nome del monaco francese Padre Marin Mersenne che fece alcune considerazioni sulla primalità di tali numeri.

Molti tra i primi scrittori pensavano che i numeri di Mersenne fossero primi se lo era l'esponente p, il che è certamente vero per i primi pochi casi.

$$2^2 - 1 = 3$$
$$2^3 - 1 = 7$$
$$2^5 - 1 = 31$$
$$2^7 - 1 = 127$$

Ma per p = 11 il numero non è primo.

$$2^{11} - 1 = 2047 = 23*89$$

Per almeno gli ultimi cento anni, il più grande numero primo conosciuto al mondo è sempre stato un numero di Mersenne (salvo un breve periodo nel 1951). Prima del Gennaio 1983, il più grande numero primo conosciuto era

$$2^{44497} - 1$$

un numero di Mersenne scoperto da David Slowinski nell'Aprile 1979. Era il 27esimo numero primo di Mersenne conosciuto. Nel Gennaio 1983 Slowinski trovò un numero primo ancora più grande

$$2^{86243} - 1$$

che è un numero di Mersenne con 25962 cifre decimali. (Ci vorrebbero molte pagine di questo libro per scrivere tutte le cifre del numero). Il calcolatore usato da Slowinski era il Cray 1 che è un macchina sorprendentemente veloce tanto che impiegò 1 ora, 36 minuti e 22 secondi di tempo macchina per eseguire la verifica.

Usando uno dei programmi del paragrafo precedente, sarebbe impossibile verificare la primalità di un numero generico della grandezza di $2^{86243} - 1$. Ma esistono tecniche speciali per i numeri di Mersenne, sviluppate negli ulti-

mi cento anni. Per verificare la primalità di numeri della forma di $N = 2^P - 1$, si può definire una serie come questa:

$U_1 = 4$
$U_2 = U_1 * U_1 - 2 - N * INT((U_1 * U_1 - 2)/N)$
.....
.....
.....
$U_{P-1} = U_{P-2} * U_{P-2} - 2 - N * INT((U_{P-2} * U_{P-2} - 2)/N)$

Allora N è primo se e solo se U =0. Questo significa che per verificare che N è primo, noi dobbiamo eseguire approssimativamente P calcoli. Eseguire 86243 operazioni sarebbe piuttosto semplice per il Commodore 64, se i numeri coinvolti fossero piccoli. Qui, invece, si ha a che fare con numeri che hanno 86243 cifre binarie ed il Commodore 64 memorizza numeri binari di sole 32 cifre: esistono però metodi per aggirare questo problema sui quali non entreremo in dettaglio qui, ma nel prossimo capitolo.

Verifica probabilistica di primalità

Avendo tempo infinito, potremmo verificare la primalità di un intero per divisioni successive, ma il tempo è limitato, anche per un calcolatore.

Negli ultimi anni è stato escogitato un metodo basato su un vecchio teorema di Pierre Fermat che visse nel 17esimo secolo. Fermat dimostrò che, se P è un numero primo e B è un altro numero compreso tra 1 e P–1, allora $B^{P-1}-1$ è divisibile per P. Ad esempio se P=11 e B=2 allora $2^{11-1}-1$ è 1023 che è certamente divisibile per 11.

Sebbene Fermat abbia dimostrato il suo teorema per tutti i valori di B, il matematico cinese Pomerance (V secolo a.C.) conosceva il teorema nel caso particolare di B=2. Inoltre pensava, sbagliando, che era vero anche il contrario e cioé che, se $2^{P-1}-1$ è divisibile per P, allora P è un numero primo. Infatti il numero 341 divide $2^{340}-1$ ma 341 non è primo e noi lo chiamiamo uno *pseudoprimo in base 2*. In generale un numero composto P che divide B^{P-1} è detto *pseudoprimo in base B* e molti numeri che sembrano essere pseudoprimi sono in realtà veri numeri primi. Ed è proprio su questo che è basata la verifica seguente.

```
10 REM TEST PROBABILISTICO DI PRIMALITA'
20 PRINTCHR$ (147)CHR$ (154)"      TEST P
```

```
ROBABILISTICO DI PRIMALITA'"CHR$ (17)
30 INPUT"NUMERO DA VERIFICARE";N
40 IFN<4ORN<>INT(N)THENPRINT"SII RAGIONE
VOLE":PRINT:GOTO30
50 REM FATTORE N-1=(2↑T)*X
60 T=0:X=N-1
70 D=X/2:IFINT(D)=DTHENT=T+1:X=D:GOTO70
80 REM SELEZIONA LA BASE B
90 B=RND(-TI):B=INT(RND(1)*(N-3.0001)+2)
100 PRINT:PRINTCHR$ (5)"TEST CHE USA LA
BASE";B;CHR$ (17)
110 REM ELEVA B ALLA POTENZA X
120 P=1
130 IFX=0THEN170
140 D=X/2:IFD<>INT(D)THENP=P*B:P=P-N*INT
(P/N)
150 B=B*B:B=B-N*INT(B/N):X=INT(D):GOTO13
0
160 REM VERIFICA B↑X
170 IFP=1ORP=N-1THEN200
180 IFT<2THENPRINT"IL NUMERO NON E' PRIM
O":GOTO250
190 P=P*P:P=P-N*INT(P/N):T=T-1:GOTO170
200 PRINT"IL NUMERO E' PROBABILMENTE PRI
MO"CHR$ (17)
210 PRINTCHR$ (158)"VUOI NUOVAMENTE VERI
FICARE IL NUMERO CON"
220 PRINT"UN'ALTRA BASE? S/N"
230 GETG$:IFG$<>"S"ANDG$<>"N"THEN230
240 IFG$="S"THEN60
250 PRINT:PRINTCHR$ (154),"    ANCORA? S/
N"
260 GETG$:IFG$<>"S"ANDG$<>"N"THEN260
270 IFG$="S"THENRUN
280 PRINTCHR$ (147),"    ARRIVEDERCI":END
```

Provate numeri come 341 e 561, nessuno dei quali è primo. Se trovate che sono pseudoprimi in una qualche base, provate con un'altra.

Nota. Il programma è veloce, ma non lavora molto bene con numeri grandi a causa degli arrotondamenti nelle moltiplicazioni.

CAPITOLO 10

MISCELLANEA

Terne Pitagoriche

Rivediamo il Teorema di Pitagora che abbiamo già ricordato in questo libro. Questo afferma che in un triangolo rettangolo con lati X, Y e Z dove Z è l'ipotenusa (il lato più lungo) si ha la seguente relazione:

$$X^2 + Y^2 = Z^2$$

Il classico esempio che viene facilmente alla mente è il triangolo rettangolo di lati 3, 4 e 5.

$$3^2 + 4^2 = 5^2$$

Un altro esempio è il triangolo rettangolo di lati 5, 12 e 13.

$$5^2 + 12^2 = 13^2$$

Naturalmente vi sono infiniti differenti esempi di triangoli rettangoli, ma quanti sono i casi in cui X, Y e Z sono interi? La risposta è che anche questi sono infiniti e tali numeri sono detti *Terne Pitagoriche.*

Come si fa a produrre un elenco di tali terne? È chiaro che tale elenco non deve contenere terne che sono multipli di altre quale 6, 8 e 10 rispetto alla Terna Pitagorica 3, 4 e 5. Le Terne Pitagoriche che non hanno fattori comuni, sono dette *primitive* e quindi 3, 4 e 5 è una Terna Pitagorica primitiva mentre 6, 8 e 10 non lo è.

Produrre Terne Pitagoriche può essere dispendioso in termini di tempo, ma non certo per alcuni matematici che si sono impegnati a produrre un metodo elegante per generare terne Pitagoriche, così come qui descritto.

1. Scegliete due interi positivi A e B tali che:

   (a) A è maggiore di B

(b) A+B è dispari (perciò uno dei numeri è dispari e l'altro pari)

(c) A e B non hanno comuni divisori eccetto 1.

2. Calcolate

$X = A^2 - B^2$
$Y = 2*A*B$
$Z = A^2 + B^2$

I numeri X, Y e Z costituiscono una Terna Pitagorica primitiva e di conseguenza qualsiasi terna può essere ottenuta in questo modo. Questa tecnica è usata nel programma Terne Pitagoriche per produrre elenchi di terne. Il procedimento inizia con A=2 e prosegue ricercando tutti i possibili valori di B che soddisfano la condizione richiesta. Il valore di A è incrementato ed il procedimento viene ripetuto a gruppi di 20 terne alla volta e potete continuare finché volete. Il valore iniziale di A può essere variato a piacimento.

```
10 REM TERNE PITAGORICHE
20 PRINTCHR$(147),CHR$(154)"TERNE PITA
GORICHE"CHR$(17)
30 PRINT"QUESTO PROGRAMMA STAMPA ALCUNE
TERNE"
40 PRINT"PITAGORICHE PRIMITIVE"CHR$(17)
50 PRINT,"PREMI S PER CONTINUARE":K=0:A=
2:B=3
60 GETG$:IFG$<>"S"THEN60
70 PRINTCHR$(147),CHR$(154)" TERNE PIT
AGORICHE"CHR$(17)
80 PRINT " NUM.    ** X **    ** Y **    **
 Z **"CHR$(158)
90 B=B-2:IFB<1THENA=A+1:B=A-1
100 A1=A:B1=B
110 N=INT(A1/B1):A2=A1-N*B1
120 IFA2<>0THENA1=B1:B1=A2:GOTO110
130 IFB1<>1THEN90
140 K=K+1:PRINTK,A*A-B*B,2*A*B,A*A+B*B
150 IFINT(K/20)<>K/20THEN90
160 PRINT:PRINTCHR$(154),"    ANCORA ? S
/N";
170 GETG$:IFG$<>"S"ANDG$<>"N"THEN170
180 IFG$="S"THEN70
190 PRINTCHR$(147),"    ARRIVEDERCI":END
```

Potenze in multi-precisione

Il Commodore 64 tratta con accuratezza numeri con 9 cifre significative e quindi sono memorizzati correttamente numeri fino a 999,999,999. Numeri più grandi sono arrotondati. Come possiamo calcolare esattamente grandi potenze di numeri? Ad esempio, quanto fa 2 elevato a 130? La soluzione è usare un'aritmetica in multi-precisione.

Un modo di produrre aritmetica in multiprecisione è di immagazzinare le cifre del numero in una matrice M(I) in cui M(0) contiene le ultime 4 cifre, M(1) contiene le 4 precedenti e così via. Perciò 987654 verrebbe immagazzinato come:

M(1) = 98 M(0) = 7654

Il numero originale è ottenuto convertendo gli elementi della matrice in stringhe e stampandole una dietro all'altra. Naturalmente ciò può essere anche ottenuto attraverso una formula:

$M = M(1) * 10^4 + M(0)$

Supponiamo di voler moltiplicare M = 987654 per N = 23456. Prima di tutto scriviamo questi numeri nelle matrici M(I) e N(I).

M(1) = 98 M(0) = 7654
N(1) = 2 N(0) = 3456

Ora, formiamo i seguenti dati:

C(0) = M(0) * N(0)
C(1) = M(1)*N(0) + M(0)*N(1)
C(2) = M(1) * N(1)

Il risultato diventa:

C(0) = 26452224
C(1) = 353996
C(2) = 196

Infine calcoliamo il prodotto M*N così:

$196*10^8 + 353996*10^4 + 26452224$
= 19600000000 + 3539960000 + 26452224
= 23166412224

Naturalmente il calcolo non è fatto così – il Commodore 64 semplicemente arrotonda i numeri. Quello che dobbiamo fare è separare le cifre di sinistra di C(0) fino a lasciarne 4 e aggiungere a C(1) quelle separate.

C(0) → 2224
C(1) → 353996 + 2645 = 356641

Poi, separiamo le cifre di sinistra di C(1) fino a lasciarne 4 e aggiungiamo a C(2) le cifre separate.

C(1) → 6641
C(2) → 196 + 35 = 231

Di conseguenza otteniamo:

C(2) = 231 C(1) = 6641 C(0) = 2224

dal che si legge immediatamente:

987654 * 23456 = 23166412224

Questo procedimento è del tutto generale. Per altri grandi numeri ci può essere un valore per gli elementi M(2), M(3) ecc. della matrice ed in tale situazione otterremo:

C(0) = M(0)*N(0)
C(1) = M(1)*N(0) + M(0)*N(1)
C(2) = M(2)*N(0) + M(1)*N(1) + M(0)*N(2)
ecc.

Dopodichè si può eseguire il processo di separazione per stampare alla fine la riposta.

Il programma Potenze in Multiprecisione illustra come il processo descritto può essere usato per calcolare accuratamente potenze di numeri. Il programma continua fino alla 40esima cifra. Se lo volete, potete avere un grado maggiore di precisione cambiando il valore di X nella linea 80. Il grado di precisione è X+1 moltiplicato 4.

```
10 REM POTENZE IN MULTIPRECISIONE
20 PRINTCHR$ (147)CHR$ (154)"        POTE
NZE IN MULTIPRECISIONE"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA STAMPA POTENZE
 DI UN"
```

```
40 PRINT"INTERO CON PRECISIONE DI 40 CIF
RE"CHR$ (17)
50 PRINT"INTRODUCI IL NUMERO DI CUI RICH
IEDI LE   POTENZE"
60 INPUT"NUMERO ";N:PRINT
70 IFN<20RN>999999990RN<>INT(N)THENPRIN
T"ERRORE - PROVA DI NUOVO":GOTO60
80 T=10000:X=9:Y=X+2:DIMM(X),N(X),L(Y):K
=1:N(0)=N:M(0)=N
85 REM CAMBIA X CON UN DIVERSO GRADO DI
PRECISIONE
90 FORI=0TO2
100 IFM(I)>=TTHENQ=INT(M(I)/T):M(I)=M(I)
-Q*T:M(I+1)=M(I+1)+Q
110 N(I)=M(I)
120 NEXT
130 PRINTCHR$ (147)"POTENZE IN MULTIPREC
ISIONE DI";N;CHR$ (17)
140 K=K+1
150 FORI=0TOY:L(I)=0:NEXT
160 FORJ=0TO2:FORI=0TOX
170 L(I+J)=L(I+J)+N(I)*M(J)
180 NEXT:NEXT
190 FORI=0TOX:N(I)=L(I):NEXT
200 FORI=0TOX
210 IFN(I)>=TTHENQ=INT(N(I)/T):N(I)=N(I)
-Q*T:N(I+1)=N(I+1)+Q
220 NEXT
230 FORI=0TOX
240 IFN(I)>0THENL=I
250 NEXT
260 PRINTCHR$ (154);N;"↑";K;CHR$ (158)
270 FORI=LTO0STEP-1
280 A$=MID$ (STR$(N(I)),2)
290 IFI<LANDLEN(A$)<>4THENA$="0"+A$:GOTO
290
300 PRINTA$;:NEXT:PRINT
310 IF(N(X)*T+N(X-1))*N>=T*TTHEN360
320 IFINT(K/10)<>K/10THEN140
330 PRINT:PRINTCHR$ (154),"   CONTINUI?
S/N";
340 GETG$:IFG$<>"S"ANDG$<>"N"THEN340
```

```
350 IFG$="S"THEN130
360 PRINT:PRINTCHR$ (154),"   ANCORA? S/
N";
370 GETG$:IFG$<>"S"ANDG$<>"N"THEN370
380 IFG$="S"THENRUN
390 PRINTCHR$ (147),"   ARRIVEDERCI":END
```

CAPITOLO 11

MATRICI

Le matrici sono tabelle rettangolari di numeri in cui conta la posizione oltre al valore di ciascun numero. Una matrice è normalmente, ma non sempre indicata entro parentesi quadre, così:

$$\begin{bmatrix} 2 & -3 & 6 \\ -4 & 4 & 5 \end{bmatrix}$$

Le dimensioni della matrice sono date dal numero di righe e di colonne e diciamo che una matrice è una matrice M*N se ha M righe e N colonne. Perciò l'esempio precedente è una matrice 2*3. Qui vi sono altri esempi di matrici

$$\begin{matrix} 8 & 12 & 0 \\ 1 & 1 & 8 \\ 8 & 12 & 1 \end{matrix} \quad \text{matrice } 3*3$$

$$\begin{matrix} 9 & 12 \\ 0 & 1 \\ 8 & -2 \end{matrix} \quad \text{matrice } 3*2$$

$$\begin{matrix} 9 & 12 \\ 3 & 2 \end{matrix} \quad \text{matrice } 2*2$$

Se in una matrice il numero di righe è uguale al numero di colonne, si dice che è una matrice quadrata.

Le matrici sono concetti utili per scrivere informazioni in forma concisa. Per esempio, supponiamo che abbiate calcolato il totale delle spese di casa trimestrali divise per tre tipi di energia: potreste visualizzare le informazioni in una tabella come questa:

Trimestre	Gas	Energia Elettrica	Nafta
1	30	28	5
2	27	19	4
3	25	15	0
4	32	27	3

I valorii possono essere messi in una matrice 4*3

$$\begin{bmatrix} 30 & 28 & 5 \\ 27 & 19 & 4 \\ 25 & 15 & 0 \\ 32 & 27 & 3 \end{bmatrix}$$

Nel vostro Commodore 64, potete memorizzare matrici usando la forma A(I,J) e quindi la matrice delle spese di energia, viste sopra, può essere memorizzata con le seguenti assegnazioni:

```
A(1,1) = 30 A(1,2) = 28 A(1,3) = 5
A(2,1) = 27 A(2,2) = 19 A(2,3) = 4
A(3,1) = 25 A(3,2) = 15 A(3,3) = 0
A(4,1) = 32 A(4,2) = 27 A(4,3) = 3
```

Le informazioni possono essere anche immagazzinate usando l'istruzione READ.

```
FOR I =1 TO4:FOR J=1 TO 3:READ A(I,J):NEXT:NEXT
DATA 30,28,5,27,19,4,25,15,0,32,27,3
```

Questo breve programma mostra come i dati possono essere introdotti e visualizzati.

```
10 PRINT CHR$(147)
20 FOR I=1 TO4:FOR J=1 TO3:READ A(I,J):NEXT:NEXT
30 FOR I=1 TO4
40 FOR J=1 TO3:PRINT A(I,J);:NEXT
50 PRINT
60 NEXT
100 DATA 30,28,5,27,19,4,25,15,0,32,27,3
```

Notate che per sfruttare al meglio la memoria del Commodore 64 avremmo dovuto far partire i contatori I e J da 0 anziché da 1. Comunque, concettualmente non vi sono problemi.

Notate anche che, se il numero di righe o colonne supera 10, è necessario usare l'istruzione DIM.

Somma di matrici

Due matrici possono essere sommate tra loro, purché abbiano le stesse dimensioni, ovvero lo stesso numero di righe e di colonne. L'operazione è ottenuta sommando termine a termine, nelle rispettive posizioni. Ad esempio

$$\begin{bmatrix} 1 & 5 & 4 \\ -2 & 3 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 5 \\ 3 & 2 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1+2 & 5+0 & 4+5 \\ -2+3 & 3+2 & 1+1 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 5 & 9 \\ 1 & 5 & 2 \end{bmatrix}$$

Perché sommare le matrici?

Abbiamo visto nel paragrafo precedente un esempio di spese trimestrali per energia e supponiamo ora che l'anno successivo si abbia una analoga matrice con i seguenti dati:

$$\begin{bmatrix} 34 & 27 & 4 \\ 30 & 20 & 3 \\ 20 & 16 & 1 \\ 35 & 29 & 5 \end{bmatrix}$$

Sommando le due matrici possiamo ottenere il totale delle spese, divise per tipo di energia, per ognuno dei trimestri del biennio. Il risultato è nella matrice seguente:

$$\begin{bmatrix} 64 & 55 & 9 \\ 57 & 39 & 7 \\ 45 & 31 & 1 \\ 67 & 56 & 8 \end{bmatrix}$$

Il vostro Commodore 64 può eseguire con facilità la somma di matrici o tabelle bidimensionali. Supponiamo di avere due matrici A(I,J) e B(I,J) dove I varia da 1 a 4 e J da 1 a 3. La somma delle due matrici ne produrrà una terza C(I,J) nel seguente modo:

```
FOR I = 1 TO4:FOR J=1 TO3
C(I,J) = A(I,J) + B(I,J)
NEXT:NEXT
```

La sottrazione di matrici segue le stesse regole.

Prodotto di matrici

Il prodotto di due matrici è un concetto un po' più complicato. Per poter eseguire il prodotto, è necessario che il numero di colonne della prima matrice sia uguale al numero di righe della seconda e quindi una matrice 4*3 può essere moltiplicata per una matrice 3*2, ma non per una matrice 2*3.

Il prodotto di una matrice M per N con una N per P sarà una matrice M per P. Lo svolgimento dei calcoli è mostrato con un esempio, mentre la spiegazione sarà data successivamente. Supponiamo di voler moltiplicare le seguenti matrici:

$$\begin{bmatrix} 2 & -3 & 6 \\ -4 & 4 & 5 \end{bmatrix}$$

Prima matrice 2*3.

$$\begin{bmatrix} 9 & 12 \\ 0 & 1 \\ 8 & -2 \end{bmatrix}$$

Seconda matrice 3*2

Il numero di colonne della prima matrice uguaglia il numero di righe della seconda ed è quindi possibile moltiplicare la prima per la seconda.

$$\begin{bmatrix} 2 & -3 & 6 \\ -4 & 4 & 5 \end{bmatrix} * \begin{bmatrix} 9 & 12 \\ 0 & 1 \\ 8 & -2 \end{bmatrix}$$

$$\begin{bmatrix} 2*9 + -3*0 + 6*8 & 2*12 + -3*1 + 6*-2 \\ -4*9 + 4*0 + 5*8 & -4*12 + 4*1 + 4*-2 \end{bmatrix}$$

$$\begin{bmatrix} 66 & 9 \\ 4 & -52 \end{bmatrix}$$

Per eseguire il prodotto, guardiamo prima di tutto ciascuna riga della prima matrice e ciascuna colonna della seconda: il numero di elementi è lo stesso. Per ciascuna di queste righe e colonne, moltiplichiamo tra loro il primo degli elementi di ognuna, poi il secondo e così via; poi sommiamo tra loro questi prodotti ottenendo un valore per la matrice-prodotto.

Con il vostro Commodore 64, si può ottenere il prodotto di una matrice A(I,J) di dimensione M*N per una matrice B(I,J) di dimensione N*P nel seguente modo, ottenendo per risultato la matrice C(I,J).

```
FOR I=1 TO M
FOR J=1 TO P
```

```
C(I,J)=0
FOR K=1TO N:C(I,J) = C(I,J)+A(I,K)*B(K,I):NEXT
NEXT
NEXT
```

Notate che se per due matrici A e B potete eseguire il prodotto A*B, allora non potete eseguire il prodotto B*A. Anche se B*A esiste, esso non è necessariamente uguale ad A*B. Cercate qualche esempio!

Perché moltiplicare le matrici?

Riprendiamo i nostri conti di casa per l'energia. Riscriviamone i dettagli.

Trimestre	Gas	Energia Elettrica	Nafta
1	30	28	5
2	27	19	4
3	25	15	0
4	32	27	3

Supponiamo che ci siano due previsioni di incremento dei prezzi.

	Previsione 1	Previsione 2
Gas	10%	5%
En. Elet.	5%	10%
Nafta	10%	10%

Gli incrementi sono dati in decimali in questa tabella

	Previsione 1	Previsione 2
Gas	0.1	0.05
En. Elet.	0.05	0.1
Nafta	0.1	0.1

Quanto di più pagherete per i tre tipi di energia in ogni trimestre? La rispo-

sta dipende da quale previsione usate e il risultato può essere evidenziato come nella tabella qui sotto.

Trimestre	Previsione 1	Previsione 2
1	30*.1+28*.05+5*.1	30*.05+28*.1+5*.1
2	27*.1+19*.05+4*.1	27*.05+19*.1+4*.1
3	25*.1+15*.05+0*.1	25*.05+15*.1+0*.1
4	32*.1+27*.05+3*.1	32*.05+27*.1+3*.1

Svolgendo i calcoli si ottiene:

Trimestre	Previsione 1	Previsione 2
1	4.9	4.8
2	4.05	3.65
3	3.24	2.75
4	4.85	4.6

Avrete forse già notato che la matrice che ne risulta è il prodotto della matrice dei costi per quella delle previsioni, cioé, in altri termini, è il prodotto delle seguenti matrici:

$$\begin{bmatrix} 30 & 28 & 5 \\ 27 & 19 & 4 \\ 25 & 15 & 0 \\ 32 & 27 & 3 \end{bmatrix} * \begin{bmatrix} 0.1 & 0.05 \\ 0.05 & 0.1 \\ 0.1 & 0.1 \end{bmatrix}$$

Zeri e uni

Le matrici che hanno tutti gli elementi nulli (zero) sono dette matrici nulle. La somma di una matrice con una nulla non ha alcun effetto, mentre il prodotto per una matrice nulla dà come risultato una matrice nulla. Per esempio:

$$\begin{bmatrix} 30 & 28 \\ 27 & 19 \\ 25 & 15 \end{bmatrix} * \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 30*0+28*0 \\ 27*0+19*0 \\ 25*0+15*0 \end{bmatrix} \quad \begin{bmatrix} 30*0+28*0 \\ 27*0+19*0 \\ 25*0+15*0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Il prodotto di due matrici non nulle può essere zero come nell'esempio qui sotto.

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1*1+1*-1 \\ 1*1+1*-1 \end{bmatrix} \qquad \begin{bmatrix} 1*-1+1*1 \\ 1*-1+1*1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Una matrice quadrata che ha 1 nei termini della diagonale (da sinistra in alto a destra in basso) e 0 in tutti gli altri, è detta matrice *identità* come sono entrambe le due matrici dell'esempio che segue.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Una matrice moltiplicata per una matrice identità è uguale a se stessa. Ovverossia la matrice identità si comporta come il numero 1 nelle normali moltiplicazioni. Ad esempio:

$$\begin{bmatrix} 30 & 28 \\ 27 & 19 \\ 25 & 15 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 30*1+28*0 \\ 27*1+19*0 \\ 25*1+15*0 \end{bmatrix} \qquad \begin{bmatrix} 30*0+28*1 \\ 27*0+19*1 \\ 25*0+15*1 \end{bmatrix}$$

$$= \begin{bmatrix} 30 & 28 \\ 27 & 19 \\ 25 & 15 \end{bmatrix}$$

Inversione di una matrice

L'inverso o reciproco di una matrice, se esiste, ha la stessa proprietà dell'inverso di un numero. Poiché l'inverso di 4 è 0.25 risulta:

$$4 * 0.25 = 1, \qquad 0.25 * 4 = 1$$

Per una matrice A, l'inverso (se esiste) è indicato con A e

$$A * A^{-1} = I, \qquad A * A^{-1} = I$$

dove I indica la matrice identità.

Solo le matrici quadrate possono avere l'inverso, ed in realtà solo alcune di esse.

Esistono diversi modo per calcolare l'inverso di una matrice: uno in particolare di tipo passo-passo si adatta molto bene al vostro Commodore 64 e consiste nell'eseguire le cosiddette 'operazioni per riga'. In termini brutali, mettiamo una matrice identità vicino alla matrice di cui vogliamo l'inverso. Le 'operazioni per riga' sono eseguite sulle due matrici contemporaneamente finché la matrice originale non è convertita in una matrice identità. Quella che prima era la matrice identità, è ora la matrice inversa.

Il programma per calcolare le matrici inverse è esposto più avanti, mentre adesso spieghiamo brevemente il metodo.

Supponiamo di voler trovare l'inverso della matrice seguente:

$$\begin{bmatrix} 2 & 1 \\ -2 & 4 \end{bmatrix}$$

Mettiamole vicino la matrice identità.

$$\begin{bmatrix} 2 & 1 \\ -2 & 4 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Vogliamo che il primo elemento in alto a sinistra sia un 1 e quindi dividiamo la prima fila per 2.

$$\begin{bmatrix} 1 & 0.5 \\ -2 & 4 \end{bmatrix} \qquad \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}$$

Poi vogliamo che l'elemento in basso a sinistra sia 0 e quindi aggiungiamo due volte la prima riga alla seconda:

$$\begin{bmatrix} 1 & 0.5 \\ 0 & 5 \end{bmatrix} \qquad \begin{bmatrix} 0.5 & 0 \\ 1 & 1 \end{bmatrix}$$

Ora vogliamo che l'elemento in basso a destra (della matrice di sinistra) sia 1 e questo lo otteniamo dividendo per 5 la seconda riga.

$$\begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0.5 & 0 \\ 0.2 & 0.2 \end{bmatrix}$$

Per far diventare la matrice di sinistra una identità, sottraiamo la metà della seconda riga alla prima.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0.4 & -0.1 \\ 0.2 & 0.2 \end{bmatrix}$$

La matrice a sinistra è l'inverso della matrice di partenza come si verifica con i calcoli seguenti.

$$\begin{bmatrix} 2 & 1 \\ -2 & 4 \end{bmatrix} * \begin{bmatrix} 0.4 & -0.1 \\ 0.2 & 0.2 \end{bmatrix}$$

$$= \begin{bmatrix} 2*.4+1*.2 & 2*-.1+1*.2 \\ -2*.4+4*.2 & -2*.1+4*.2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Il programma che segue calcola l'inverso di una matrice, sempreché questo esista.

```
10 REM INVERSIONE DI MATRICI
20 PRINTCHR$ (147),CHR$ (154)"INVERSIONE
 DI MATRICI"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA CALCOLA L'INVE
RSO DI   UNA MATRICE N PER N."CHR$ (17)
40 PRINT"SCRIVETE LA DIMENSIONE N DELLA
MATRICE  QUADRATA"CHR$ (17)
50 INPUT"VALORE DI N";N:PRINT
60 IFN<1ORN<>INT(N)THENPRINT"ERRORE!! PR
OVA DI NUOVO":GOTO50
70 PRINT"INTRODUCI LA MATRICE UN TERMINE
 ALLA VOLTA"CHR$ (17)
80 DIMA(N,N),B(N,N)
90 REM LEGGI LA MATRICE
100 FORI=1TON
110 PRINTCHR$ (159);"  RIGA";I;CHR$ (154
)
```

```
120 FORJ=1TON
130 PRINT"COLONNA";J;:INPUTA(I,J)
140 NEXT:PRINT:NEXT
150 REM CALCOLO
160 PRINTCHR$ (147)"LA MATRICE:"CHR$ (17
)
170 FORI=1TON:FORJ=1TON:PRINTA(I,J);:NEX
T:PRINT:NEXT
180 PRINT:PRINT"CALCOLO IN CORSO";
190 FORI=1TON:B(I,I)=1:NEXT
200 X=0:GOSUB300
210 REM CHIUSURA
220 PRINT:PRINTCHR$ (154),"   ANCORA? S/
N"
230 GETG$:IFG$<>"S"ANDG$<>"N"THEN230
240 IFG$="S"THENRUN
250 PRINTCHR$ (147),"    ARRIVEDERI":END
300 REM CALCOLO
310 X=X+1:Z=X:PRINT"*";
320 IFA(Z,X)=0THENZ=Z+1:IFZ<=NTHEN320
330 IFZ>NTHENPRINT:PRINT:PRINT"NON INVER
TIBILE!!":RETURN
340 IFZ<>XTHENR=1/A(Z,X):I=X:K=Z:GOSUB50
0
350 IFA(X,X)<>1THENR=1/A(X,X):I=X:GOSUB5
50
360 FORI=1TON
370 IFI=XTHENI=I+1:IFI>NTHEN390
380 IFA(I,X)<>0THENR=-A(I,X):K=X:GOSUB50
0
390 NEXT
400 IFX<NTHEN310
410 PRINT:PRINT:PRINT"L'INVERSO:"CHR$ (1
7)
420 FORI=1TON:FORJ=1TON:PRINTB(I,J);:NEX
T:PRINT:NEXT
430 RETURN
500 REM OPERAZIONE SULLE RIGHE - SOMMA R
 VOLTE LA RIGA K ALLA RIGA I
510 FORJ=1TON
520 A(I,J)=A(I,J)+R*A(K,J):B(I,J)=B(I,J)
+R*B(K,J)
```

```
530 NEXT
540 RETURN
550 REM OPERAZIONE SULLE RIGHE MOLTIPLIC
A LA RIGA I PER R
560 FORJ=1TON
570 A(I,J)=R*A(I,J):B(I,J)=R*B(I,J)
580 NEXT
590 RETURN
```

Sistemi di equazioni

Le matrici sono utili nelle risoluzioni di sistemi di equazioni. Un esempio di un sistema di 2 equazioni a 2 incognite è il seguente.

$$3*x + 1*y = 7$$
$$5*x + 2*y = 9$$

Che, in forma matriciale, è scritto così:

$$\begin{bmatrix} 3 & 1 \\ 5 & 2 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 7 \\ 9 \end{bmatrix}$$

Un metodo per risolvere il problema è di trovare l'inverso della matrice 2*2 di sinistra e poi moltiplicare l'equazione per il suo inverso. Nell'esempio la matrice inversa è:

$$\begin{bmatrix} 2 & -1 \\ -5 & 3 \end{bmatrix}$$

Perciò otteniamo le seguenti equazioni:

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix}$$

$$= \begin{bmatrix} 2 & -1 \\ -5 & 3 \end{bmatrix} * \begin{bmatrix} 3 & 1 \\ 5 & 2 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix}$$

$$= \begin{bmatrix} 2 & -1 \\ -5 & 3 \end{bmatrix} * \begin{bmatrix} 7 \\ 9 \end{bmatrix}$$

$$= \begin{bmatrix} 2*7-1*9 \\ -5*7+3*9 \end{bmatrix}$$

$$= \begin{bmatrix} 5 \\ -8 \end{bmatrix}$$

Quindi x=5 e y=−8 risolvono il sistema di equazioni.

Il programma che segue, Sistemi di Equazioni, in sostanza segue questo procedimento per risolvere sistemi di N equazioni in N incognite. In realtà, si tratta di scrivere il sistema in due matrici A(I,J) e B(I). L'esempio precedente verrebbe scritto così:

$$\begin{bmatrix} 3 & 1 \\ 5 & 2 \end{bmatrix} \qquad \begin{bmatrix} 7 \\ 9 \end{bmatrix}$$

Si eseguono le operazioni sulle righe finché la matrice di sinistra diventa una matrice identità. A quel punto, la matrice di destra contiene la soluzione del sistema di equazioni.

```
10 REM SISTEMI DI EQUAZIONI
20 PRINTCHR$ (147),CHR$ (154)"SISTEMI DI
 EQUAZIONI"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA RISOLVE UN SIS
TEMA DI"
40 PRINT"N EQUAZIONI IN N INCOGNITE"CHR$
 (17)
50 PRINT"INTRODUCI IL NUMERO N DI EQUAZI
ONI"CHR$ (17)
60 INPUT"VALORE DI N";N:PRINT
70 IFN<1ORN<>INT(N)THENPRINT"ERRORE!! PR
OVA DI NUOVO":GOTO60
80 PRINT"INTRODUCI I COEFFICIENTI UN TER
MINE ALLA VOLTA"CHR$ (17)
90 DIMA(N,N),B(N)
100 REM ACCETTA LA MATRICE DEI COEFFICIE
NTI
```

```
110 FORI=1TON
120 PRINTCHR$ (159);"  RIGA";I;CHR$ (154
)
130 FORJ=1TON
140 PRINT"COLONNA";J;:INPUTA(I,J)
150 NEXT
160 INPUT"TERMINE DI DESTRA";B(I):PRINT
170 NEXT
180 REM CALCOLO
190 PRINTCHR$ (147)"LE MATRICI DEI COEFF
ICIENTI:"CHR$ (17)
200 FORI=1TON:FORJ=1TON:PRINTA(I,J);:NEX
T:PRINT,B(I):NEXT
210 PRINT:PRINT"CALCOLO INCORSO";
220 X=0:GOSUB300
230 REM CHIUSURA
240 PRINT:PRINT:PRINTCHR$ (154),"   ANCO
RA? S/N"
250 GETG$:IFG$<>"S"ANDG$<>"N"THEN250
260 IFG$="S"THENRUN
270 PRINTCHR$ (147),"   ARRIVEDERCI":END
300 REM CALCOLO
310 X=X+1:Z=X:PRINT"*";
320 IFA(Z,X)=0THENZ=Z+1:IFZ<=NTHEN320
330 IFZ>NTHENPRINT:PRINT:PRINT"SOLUZIONE
 IMPOSSIBILE!!":RETURN
340 IFZ<>XTHENR=1/A(Z,X):I=X:K=Z:GOSUB50
0
350 IFA(X,X)<>1THENR=1/A(X,X):I=X:GOSUB5
50
360 FORI=1TON
370 IFI=XTHENI=I+1:IFI>NTHEN390
380 IFA(I,X)<>0THENR=-A(I,X):K=X:GOSUB50
0
390 NEXT
400 IFX<NTHEN310
410 PRINT:PRINT:PRINT"LA SOLUZIONE:"CHR$
 (17)
420 FORI=1TON
430 IFPOS(1)+LEN(STR$(B(I)))>38THENPRINT
440 PRINTB(I);
450 NEXT
```

```
460 PRINT:RETURN
500 REM OPERAZIONE SULLE RIGHE - SOMMA R
 VOLTE LA RIGA K ALLA RIGA I
510 FORJ=1TON
520 A(I,J)=A(I,J)+R*A(K,J)
530 NEXT
540 B(I)=B(I)+R*B(K):RETURN
550 REM OPERAZIONE SULLE RIGHE MOLTIPLIC
A LA RIGA I PER R
560 FORJ=1TON
570 A(I,J)=R*A(I,J)
580 NEXT
590 B(I)=R*B(I):RETURN
```

CAPITOLO 12

CODICI

La crescente necessità di garantire sicure trasmissioni di dati ha fatto sviluppare le ricerche di crittografia – l'arte di scrivere in codice o in cifre –. Nella crittografia vi sono due passi fondamentali, prima la codifica dei messaggi o dati e poi la decodifica di quanto ricevuto per ricostruire l'originale.

Codici a sostituzione

La cifratura più semplice è quella a sostituzione in cui ogni lettera è sostituita da qualcosa d'altro, normalmente una lettera dello stesso alfabeto. Eccone un esempio:

Codifica→ ←decodifica	codifica→ ←decodifica
A – A	B – F
C – K	D – P
E – U	F – Z
G – E	H – J
I – O	J – T
K – Y	L – D
M – I	N – N
O – S	P – X
Q – C	R – H
S – M	T – R
U – W	V – B
W – G	X – L
Y – Q	Z – V

La parola MESSAGE verrebbe tradotta in IUMMAEU.

In questo codice, la sostituzione di ogni lettera non è scelta a caso. Se le lettere dell'alfabeto (inglese) sono numerate da 0 a 25, allora la sostituzione per la lettera di numero NUM si calcola così:

```
NUM*5 - 26*INT(NUM*5/26)
```

ed il processo inverso è ottenuto con

```
NUM*21 - 26*INT(NUM*21/26)
```

Il codice può essere elencato con il seguente programmino.

```
10 REM CIFRATURA SEMPLICE
20 PRINT CHR$(147),"CODIFICA"CHR$(17)
30 FOR I=0 TO 25
40 J=I*5-26*INT(I*5/26)
50 PRINT CHR$(65+I)"-"CHR$(65+J),
60 NEXT
70 PRINT:PRINT:PRINT,"DECODIFICA"CHR$(17)
80 FOR I=0 TO25
100 J=I*21-26*INT(I*21/26)
110 PRINT CHR$(65+I)"-"CHR$(65+J),
120 NEXT
```

In pratica per codificare una lettera, ne moltiplichiamo il numero corrispondente per 5 ignorando i multipli di 26 mentre, per decodificarla, dobbiamo dividere per 5 ignorando ancora i multipli di 26. Questo è lo stesso che moltiplicare per 21 ignorando i multipli di 26 perché

```
5*21 = 105
     = 1+4*26
```

In altre parole, trascurando i multipli di 26, allora 21 è il reciproco di 5 ed infatti si dice che 21 è l'inverso di 5 modulo 26.

Un evidente difetto di questo codice è che un messaggio come PLEASE COME QUICKLY (in italiano = prego vieni subito) sarebbe codificato in XDUAMU KSIU CWIKYDQ con gli spazi che rimangono tali. Per superare questo scoglio nel nostro elenco di sostituzioni, potremmo includere gli spazi, i punti, le virgole, i numeri e fors'anche il punto di domanda.

Avendo a disposizione un calcolatore, potremmo sfruttare i caratteri ASCII, in particolare i 59 compresi tra i valori ASCII 32 e 90. Questi includono tutte le lettere che ci servono e oltretutto 59 è un numero primo che sarà utile per i nostri scopi.

La codifica viene fatta, essenzialmente, moltiplicando per 5 e, più in dettaglio, il carattere A$ sarà codificato in C$ così.

```
N = 5*(ASC($)-32)
C$ = CHR$(32+N-59*INT(N/59))
```

La decodifica è ottenuta moltiplicando per 12 (il prodotto di 5 e 12 è 60, cioè 1 ignorando i multipli di 59) e procediamo così:

```
M = 12*(ASC(C$)-32)
A$ = CHR$(32+M-59*INT(M/59))
```

Qui viene dato un breve programma che, basandosi sul codice appena visto, può codificare e decodificare un messaggio. In più vi è un'aggiunta perché è chiesto di dare un numero tra 2 e 58 che verrà usato sia per la codifica che per la decodifica. In pratica, nel primo caso, moltiplichiamo per il numero scelto, diciamo N, mentre nell'altro moltiplichiamo per il reciproco di N modulo 59 cioé per un numero M per cui N*M è 1 trascurando i multipli di 59.

```
10 REM CODICE A SOSTITUZIONE
20 PRINTCHR$ (147),CHR$ (154)"CODICE A S
OSTITUZIONE"CHR$ (17)
30 PRINT "QUESTO PROGRAMMA PUO' CODIFICA
RE E DECO-DIFICARE.      "
40 PRINT"INTRODUCI UN NUMERO PER LA CODI
FICA"CHR$ (17)
50 INPUT"NUMERO DA 2 A 58 ";C:PRINT
60 IFC<>INT(C)ORC<2ORC>58THENPRINT"PROVA
 ANCORA.":GOTO50
70 PRINT"VUOI CODIFICARE (C) O DECODIFIC
ARE (D)  UN MESSAGGIO?"CHR$ (17)
80 INPUT"C O D";A$:PRINT
90 IFA$<>"C"ANDA$<>"D"THEN PRINT"QUALE?"
:GOTO80
100 D=1:I=C:B$="DE":IFA$="C" THEN D=C:B$
=="":GOTO130
110 C=C+I:D=D+1:IFC-59*INT(C/59)<>1THEN1
10
120 REM D E' IL MULTIPLO RICHIESTO PER L
```

```
A CODIFICA/DECODIFICA
130 PRINT "SCRIVI IL MESSAGGIO - MAX 255
 CARATTERI"CHR$ (17)
140 M$="":N$="":L=0
150 GETG$:IFG$=""THEN150
160 REM IL MESSAGGIO E' INSERITO CARATTE
RE PER CARATTERE
170 G=ASC (G$)
180 IFG>31ANDG<91THENM$=M$+G$:PRINTG$;:L
=L+1
190 IFG=20ANDL>0THENPRINTG$;:L=L-1:M$=LE
FT$ (M$,L)
200 IFG<>13ANDL<255THEN150
210 PRINT" "
220 REM MESSAGGIO DA CODIFICARE/DECODIFI
CARE
230 FORI=1TOL
240 N=ASC (MID$ (M$,I,1))-32:N=D*N
250 N$=N$+CHR$ (32+N-59*INT(N/59))
260 NEXT
270 PRINT:PRINT"IL MESSAGGIO "B$"CODIFIC
ATO E':"CHR$ (17):PRINTN$
280 PRINT:PRINTCHR$ (154),"      ANCORA?
S/N"
290 GETG$:IFG$<>"S"ANDG$<>"N"THEN290
300 IFG$="S"THENRUN
310 PRINTCHR$ (147),"      ARRIVEDERCI":E
ND
```

A dispetto dei nostri sforzi, il codice è facile (per gli esperti) da rompere o decifrare. Il problema di un codice a sostituzione è che ogni carattere è invariabilmente rappresentato da un altro ben preciso carattere. Nelle lingue alcune lettere o gruppi di lettere sono utilizzate molto più frequentemente di altre. Ad esempio, in un normale testo inglese, la lettera E compare circa 13 volte su cento, la lettera T 9 volte, la lettera P 2 volte, e Q circa 0,2 volte. Muniti di queste informazioni è possibile decifrare un codice a sostituzione. Il numero di codici ottenibili in questo modo usando i 59 caratteri (quelli ASCII compresi tra 32 e 90) è di $1.4*10^{80}$ e, ciononostante, essi possono essere facilmente decifrati. Questo fatto illustra quanto possa trarre in inganno la presenza di un elevato numero di combinazioni.

Il paragrafo che segue illustra dei codici che non usano sempre lo stesso carattere per codificarne un altro.

Codici a matrice

Per cifrare i messaggi si possono usare matrici come è illustrato in questo esempio. Posto di voler codificare il messaggio PLEASE COME QUICKLY, riscriviamolo su due file come qui sotto:

```
P E S   O E Q I K Y
L A E C M   U C L .
```

Ora creiamo una matrice convertendo le lettere nel loro codice ASCII diminuito di 31.

```
49 38 52  1 48 38 50 42 44 58
45 34 38 36 46  1 54 36 45 15
```

Adesso moltiplichiamo per la seguente matrice.

$$\begin{bmatrix} 2 & -1 \\ -5 & 3 \end{bmatrix}$$

Il risultato è:

$$\begin{bmatrix} 2 & -1 \\ -5 & 3 \end{bmatrix} * \begin{bmatrix} 49 & 38 & 52 & 1 & 48 & 38 & 50 & 42 & 44 & 58 \\ 45 & 34 & 38 & 36 & 46 & 1 & 54 & 36 & 45 & 15 \end{bmatrix}$$

$$= \begin{bmatrix} 53 & 42 & 66 & -34 & 50 & 75 & 46 & 48 & 43 & 101 \\ -110 & -88 & -146 & 103 & -102 & -187 & -88 & -102 & -85 & -245 \end{bmatrix}$$

Quindi convertiamo i numeri della matrice per far si che siano compresi tra 0 e 58. Per ottenerlo, aggiungiamo o sottraiamo multipli di 59 ai valori presenti con la seguente formula:

```
N = N - 59*INT(N/59)
```

Questa è la matrice che risulta.

$$\begin{bmatrix} 53 & 42 & 7 & 25 & 50 & 16 & 46 & 48 & 43 & 42 \\ 8 & 30 & 31 & 44 & 16 & 49 & 30 & 16 & 33 & 50 \end{bmatrix}$$

Alla fine aggiungiamo 31 a questi numeri e controlliamo i caratteri corrispondenti utilizzando PRINT CHR$(X): il risultato è qui sotto:

```
T  I  &  8  Q  /  M  O  J  I
´  =  >  K  /  P  =  /  @  Q
```

Il messaggio finale è:

T´I=& >8KQ//PM=O/J@IQ

Si noti che la lettera E compare 3 volte nel messaggio iniziale, ma è codificata con I, > e / e quindi questo cifrario è molto più difficile che non il metodo di sostituzione diretta.

Un codice però è inutile se non si può decifrare il messaggio. Il problema è adesso di invertire il procedimento e, a prima vista, questo può sembrare difficile, ma un po' di matematica ci può aiutare.

Poniamo di voler decifrare il messaggio:

I=M?"S?$:8M @ >´<M

Prima di tutto scriviamolo su due file.

```
I  M  "  ?  :  M  >  <
=  ?  S  $  8  @  ´  M
```

Poi calcoliamo i codici ASCII meno 31 per ottenere una matrice.

$$\begin{bmatrix} 42 & 46 & 3 & 32 & 27 & 46 & 31 & 29 \\ 30 & 32 & 52 & 5 & 25 & 33 & 8 & 46 \end{bmatrix}$$

Moltiplichiamo ora per la seguente matrice.

$$\begin{bmatrix} 3 & 1 \\ 5 & 2 \end{bmatrix}$$

Si noti che questa non è la stessa matrice usata per la codifica. Comunque notate anche che essa è l'inverso della matrice di codifica, come mostrato dai seguenti calcoli.

$$\begin{bmatrix} 3 & 1 \\ 5 & 2 \end{bmatrix} * \begin{bmatrix} 2 & -1 \\ -5 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} 3*2 + 1*-5 & 3*-1 + 1*3 \\ 5*2 + 2*-5 & 5*-1 + 2*3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Eseguiamo perciò la moltiplica di questa matrice di decodifica. Per la matrice del messaggio.

$$\begin{bmatrix} 3 & 1 \\ 5 & 2 \end{bmatrix} * \begin{bmatrix} 42 & 46 & 3 & 32 & 27 & 46 & 31 & 29 \\ 30 & 32 & 52 & 5 & 25 & 33 & 8 & 46 \end{bmatrix}$$

$$= \begin{bmatrix} 156 & 170 & 61 & 101 & 106 & 171 & 101 & 133 \\ 270 & 294 & 119 & 170 & 185 & 296 & 171 & 237 \end{bmatrix}$$

Ora, aggiungiamo (o sottraiamo) multipli di 59 per ottenere numeri compresi tra 0 e 58.

$$\begin{bmatrix} 38 & 52 & 2 & 42 & 47 & 53 & 42 & 15 \\ 34 & 58 & 1 & 52 & 8 & 1 & 53 & 1 \end{bmatrix}$$

Aggiungiamo ancora 31 a questi valori e guardiamone i caratteri corrispondenti.

E S ! I N T I.
A Y S T

Concludiamo con il messaggio in chiaro EASY! ISN'T IT.(=Facile vero!)

Il programma che segue usa tali matrici per cifrare e decifrare i messaggi.

```
10 REM CODICE A MATRICE
20 PRINTCHR$(147),CHR$(154) "  CODICE
A MATRICE"CHR$(17)
30 PRINT"QUESTO PROGRAMMA PUO' CODIFICAR
E E DECO-DIFICARE"CHR$(17)
40 S=2:FORI=1TOS:FORJ=1TOS:READA(I,J):NE
XT:NEXT
```

```
50 PRINT"VUOI CODIFICARE (C) O DECODIFIC
ARE (D)  UN MESSAGGIO?"CHR$ (17)
60 INPUT"C O D";A$:PRINT
70 IFA$<>"C"ANDA$<>"D"THENPRINT"QUALE?":
GOTO60
80 B$=""
90 IFA$="D"THENB$="DE":FORI=1TO5:FORJ=1T
O5:READA(I,J):NEXT:NEXT
100 PRINT"SCRIVI IL MESSAGGIO-MAX 255 CA
RATTERI"CHR$ (17)
110 M$="":L=0
120 GETG$:IFG$=""THEN120
130 G=ASC (G$)
140 IFG>31ANDG<91THENM$=M$+G$:PRINTG$;:L
=L+1
150 IFG=20ANDL>0THENPRINTG$;:L=L-1:M$=LE
FT$ (M$,L)
160 IFG<>13ANDL<255THEN120
170 PRINT" ":M=INT(L/5+0.9):DIMB(5,M),C(
5,M)
180 REM CODIFICA/DECODIFICA DEL MESSAGGI
O
190 FORI=1TO5:FORJ=1TOM
200 K=5*J+I-5
210 IFK<=LTHENN=ASC (MID$ (M$,K,1))-32:B
(I,J)=N-59*INT(N/59)
220 NEXT:NEXT
230 FORI=2TO5:IFB(I,M)<1THENB(I,M)=1
240 NEXT
250 FOR I=1TO5:FORJ=1TOM
260 FORK=1TO5:C(I,J)=C(I,J)+A(I,K)*B(K,J
):NEXT
270 C(I,J)=C(I,J)-59*INT(C(I,J)/59)
280 NEXT:NEXT
290 PRINT:PRINT"IL MESSAGGIO "B$"CODIFIC
ATO E':"CHR$ (17)
300 FORJ=1TOM:FORI=1TO5
310 PRINTCHR$ (32+C(I,J));
320 NEXT:NEXT:PRINT
330 PRINT:PRINTCHR$ (154),"   ANCORA? S/
N"
340 GETG$:IFG$<>"S"ANDG$<>"N"THEN340
```

```
350 IFG$="S"THENRUN
360 PRINTCHR$ (147),"     ARRIVEDERCI":EN
D
400 DATA2,-1,-5,3
410 DATA3,1,5,2
```

Per questo cifrario abbiamo usato una matrice 2 per 2 per codificare ed il suo inverso per decodificare. In generale si può usare una qualsiasi matrice purchè abbia solo interi nei suoi termini. Questa qui è un'altra matrice che si potrebbe usare per la cifratura.

$$\begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix}$$

La corrispondente matrice di decifratura è:

$$\begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}$$

Questo metodo di codifica può essere reso più complicato usando una matrice 3 per 3 che richiede di scrivere il messaggio su tre file procedendo poi con il criterio già esposto. Qui di seguito vi sono due matrici che possono essere usate come codificatori e decodificatori.

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 1 & 3 \\ 4 & 2 & 5 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & -1 \\ -2 & -9 & 5 \\ 0 & 2 & -1 \end{bmatrix}$$

Il programma seguente usa tali matrici per cifrare e/o decifrare un messaggio. Potete inserire una vostra coppia di matrici, ma controllate bene che siano l'inverso una dell'altra e contengano solo interi.

```
10 REM CIFRARIO CON MATRICE 3*3
20 PRINTCHR$ (147),CHR$ (154)"  CIFRARIO
 A MATRICE"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA PUO' CODIFICAR
E E DECO-DIFICARE."CHR$ (17)
40 S=3:FORI=1TOS:FORJ=1TOS:READA(I,J):NE
XT:NEXT
50 PRINT"VUOI CODIFICARE (C) O DECODIFIC
ARE (D)  UN MESSAGGIO?"CHR$ (17)
60 INPUT"C O D";A$:PRINT
70 IFA$<>"C"ANDA$<>"D"THENPRINT"QUALE?":
```

```
GOTO60
80 B$=""
90 IFA$="D"THENB$="DE":FORI=1TOS:FORJ=1T
OS:READA(I,J):NEXT:NEXT
100 PRINT"SCRIVI IL MESSAGGIO-MAX 255 CA
RATTERI."CHR$ (17)
110 M$="":L=0
120 GETG$:IFG$=""THEN120
130 G=ASC (G$)
140 IFG>31ANDG<91THENM$=M$+G$:PRINTG$;:L
=L+1
150 IFG=20ANDL>0THENPRINTG$;:L=L-1:M$=LE
FT$ (M$,L)
160 IFG<>13ANDL<255THEN120
170 PRINT" ":M=INT(L/S+0.9):DIMB(S,M),C(
S,M)
180 REM MESSAGGIO DA CODIFICARE O DECODI
FICARE
190 FORI=1TOS:FORJ=1TOM
200 K=S*J+I-S
210 IFK<=LTHENN=ASC (MID$ (M$,K,1))-32:B
(I,J)=N-59*INT(N/59)
220 NEXT:NEXT
230 FORI=2TOS:IFB(I,M)<1THENB(I,M)=1
240 NEXT
250 FORI=1TOS:FORJ=1TOM
260 FORK=1TOS:C(I,J)=C(I,J)+A(I,K)*B(K,J
):NEXT
270 C(I,J)=C(I,J)-59*INT(C(I,J)/59)
280 NEXT:NEXT
290 PRINT:PRINT"IL MESSAGGIO "B$"CIFRATO
 E':"CHR$ (17)
300 FORJ=1TOM:FORI=1TOS
310 PRINTCHR$ (32+C(I,J));
320 NEXT:NEXT:PRINT
330 PRINT:PRINTCHR$ (154),"   ANCORA? S/
N"
340 GETG$:IFG$<>"S"ANDG$<>"N"THEN340
350 IFG$="S"THENRUN
360 PRINTCHR$ (147),"   ARRIVEDERCI":END
400 DATA1,0,-1,2,1,3,4,2,5
410 DATA1,2,-1,-2,-9,5,0,2,-1
```

Codici con chiavi pubbliche

I codici descritti nei paragrafi precedenti hanno un difetto. Una volta che conoscete come codificare un testo, sapete anche come decifrarlo. I criptosistemi a chiave pubblica sono differenti. Essi sono costituiti da due parti: la chiave di codifica che è resa pubblica, permettendo a chiunque di codificare il messaggio; e la chiave di decodifica che, invece, è tenuta segreta permettendo solo a chi ha sviluppato il cifrario di eseguire la decodifica.

Descriviamo ora un sistema a chiave pubblica. Cerchiamo, innanzitutto, due numeri primi P e Q molto grandi, ciascuno con almeno 50 cifre decimali rendendo così improponibile questo metodo per il vostro Commodore 64.

Poniamo che N sia il prodotto di P per Q e scegliamo un intero A minore di N ma che non abbia fattori comuni con (P−1)*(Q−1). Ora potete divulgare i numeri N ed A.

Come viene codificato il messaggio con i numeri N ed A?

Si prosegue così:

1. Traducete il messaggio in numeri (spazio=01, A=34 etc.); ora il messaggio è un numero molto grande.
2. Prendete il messaggio in forma numerica e dividetelo in blocchi di lunghezza opportuna.
3. Codificate ogni blocco B come segue:

   C = B↑A − N*INT((B↑A)/N)

   il numero B è elevato alla potenza A e i multipli di N sono sottratti per ottenere numeri compresi tra 0 e N.

Come si decodifica il messaggio risultante? Poiché il massimo comun divisore di A e (P−1)*(Q−1) è 1, possiamo trovare due numeri X eY per cui A*X + (P−1)*(Q−1)*Y = 1.

Usando X possiamo decodificare il messaggio così:

1. Dividiamo il messaggio in blocchi.
2. Per ogni blocco eseguiamo il seguente calcolo:

   C↑X−N*INT((C↑X)/N).

3. Riuniamo i blocchi in un numero solo e convertiamo le cifre in caratteri.

Il procedimento funziona perché

$$
\begin{aligned}
(B\uparrow A)\uparrow X &= B\uparrow(A*X) \\
&= B\uparrow(1-(P-1)*(Q-1)*Y) \\
&= B + \text{multipli di } N
\end{aligned}
$$

L'ultima relazione deriva da un Teorema dimostrato dal matematico Fermat.

Perché è difficile rompere questo cifrario? Notate che, per decodificare il messaggio, dobbiamo conoscere il valore di X e questo può essere ottenuto dal valore di (P−1)*(Q−1). Per conoscere però (P−1)*(Q−1), dobbiamo conoscere P e Q. Solo A e N sono stati resi pubblici. In teoria, potremmo fattorizzare N per ottenere P e Q, ma, poiché N è un numero di circa 100 cifre decimali, la fattorizzazione di un tale numero richiede una quantità enorme di tempo di un calcolatore (diversi milioni di anni). È su questa base che il cifrario è sicuro.

Quello che vi rimane da fare per produrre un codice sicuro è scrivere un programma per il Commodore 64, che sia capace di trattare numeri molto grandi. Traete spunto dal capitolo Miscellanea.

CAPITOLO 13

CASUALITÀ!

Testa o croce

Lanciando una moneta, vi è la stessa probabilità che mostri una faccia (testa) o l'altra (croce), almeno se è una moneta normale. Il vostro Commodore 64 può simulare il lancio della moneta usando la funzione interna RND.

Per un qualsiasi numero positivo X, la funzione RND(X) fornisce un numero (pseudo) casuale nell'intervallo 0 — 1 (escluso 1). Se accendete il vostro Commodore 64 e scrivete

```
FOR I = 1 TO 5:PRINT RND(1):NEXT I
```

otterrete la sequenza di numeri qui riportata:

```
.185564016
.0468986348
.827743801
.554749226
.897233831
```

Se spegnete e riaccendete il vostro 64 (non è una operazione consigliabile) e ripetete l'istruzione

```
FOR I = 1 TO 5:PRINT RND(1):NEXT I
```

otterrete la stessa sequenza di numeri. Per superare questo scoglio, di norma si fa partire un programma che richiede numeri casuali con una linea tipo

```
Y = RND(-TI)
```

che ha l'effetto di far partire una nuova sequenza di numeri casuali. In generale, se X è negativo, RND (X) fa partire una nuova sequenza di numeri casuali mentre, se X é 0, il numero che risulta è lo stesso della serie precedente.

Il seguente breve programma simula il lancio della moneta. Viene stampato un elenco che indica se è apparsa testa (T) o croce (C). Sono stampate 100 lettere (cioè 100 lanci) e viene fatto il totale delle teste e croci apparse.

```
10 REM TESTA E CROCE
20 PRINTCHR$ (147),"     TESTA E CROCE"CH
R$ (17)
30 PRINT"QUESTO PROGRAMMA SIMULA 100 LAN
CI DI UNA MONETA",CHR$ (17)
40 PRINT"PREMERE S PER CONTINUARE":K=0
50 GETG$:IFG$<>"S"THEN50
60 Y=RND(-TI)
70 K=K+1:J=0:PRINTCHR$ (147)," PROVA NU
MERO"STR$(K)"       "CHR$ (17)
80 FORI=1TO100
90 A$="T":IFRND(1)>=.5THENA$="C":J=J+1
100 PRINTA$;
110 NEXT
120 PRINT:PRINT:PRINT"NUMERO DI TESTE";1
00-J;"CROCI";J
130 PRINT:PRINTCHR$ (154),"ANCORA? S/N"
140 GETG$:IFG$<>"S"ANDG$<>"N"THEN140
150 IFG$="S"THEN60
160 PRINTCHR$ (147)"ARRIVEDERCI":END
```

Cambiate la linea 60 con:

```
60 Y = RND(-1)
```

Noterete che, ogni volta che fate partire il programma, compare la stessa sequenza di teste e croci. Questo è il motivo per cui usiamo

```
60 Y = RND(-TI)
```

per generare una sequenza di numeri casuali.

Uomini e dadi

Se un comune dado a sei facce viene lanciato, compare uno dei numeri 1, 2, 3, 4, 5, e 6 nessuno più frequentemente dell'altro. Ancora una volta, il Commodore 64 può essere usato per simulare il lancio dei dadi. Il seguente programma illustra il risultato di 120 lanci del dado, stampando alla fine il totale delle volte in cui è apparso ogni numero.

```
10 REM LANCIO DI UN DADO
20 PRINTCHR$ (147)," LANCIO DI UN DADO"C
HR$ (17)
30 PRINT" IL PROGRAMMA SIMULA 100 LANCI
DI UN      NORMALE DADO"CHR$ (17)
40 PRINT"PREMERE S PER CONTINUARE":K=0
50 GETG$:IFG$<>"S"THEN50
60 Y=RND(-TI)
70 K=K+1:FORI=1TO6:A(I)=0:NEXT
80 PRINTCHR$ (147),"     PROVA NUMERO"STR
$(K)CHR$ (17)
90 FORI=0TO100
100 L=INT(RND(1)*6)+1:PRINTSTR$(L);:A(L)
=A(L)+1
110 NEXT
120 PRINT:PRINT:FORI=1TO6:PRINT"NUMERO D
I";I;"ESTRATTI";A(I):NEXT
130 PRINT:PRINTCHR$ (154)"ANCORA? S/N"
140 GETG$:IFG$<>"S"ANDG$<>"N"THEN140
150 IFG$="S"THEN60
160 PRINTCHR$ (147),"      ARRIVEDERCI":E
ND
```

Ecco un'altra versione del programma che include la visualizzazione della faccia del dado. Il disegno delle facce è contenuto nella tabella D$(I,J).

```
10 REM DADO DISEGNATO
20 PRINTCHR$ (147)"      LANCIO DI UN DAD
O CON DISEGNO"CHR$ (17)
30 PRINT" IL PROGRAMMA SIMULA 100 LANCI
DI UN      NORMALE DADO"CHR$ (17)
40 FORI=0TO6:FORJ=0TO6:READA:A$(I)=A$(I)
+CHR$ (A):NEXT:NEXT
50 FORI=1TO6:FORJ=0TO6:READA:D$(I,J)=A$(
A):NEXT:NEXT
60 PRINT"PREMERE S PER CONTINUARE":K=0
70 GETG$:IFG$<>"S"THEN70
80 Y=RND(-TI)
90 K=K+1:PRINTCHR$ (147)," PROVA NUMERO
"STR$(K),CHR$ (17)
100 L=INT(RND(1)*6)+1:A(L)=A(L)+1
110 FORJ=0TO7:PRINTTAB(16)D$(L,J):NEXT
```

```
120 PRINT:PRINT:PRINT"CONTO DELLE ESTRAZ
IONI PER NUMERO"CHR$ (17)
125 FORI=1TO3:PRINT"  NUM"STR$(I);":"A(I
);:NEXT
130 PRINT:FORI=4TO6:PRINT"  NUM"STR$(I);
":"A(I);:NEXT:PRINT
140 PRINT:PRINTCHR$ (154)"ANCORA? S/N"
150 GETG$:IFG$<>"S"ANDG$<>"N"THEN150
160 IFG$="S"THEN80
170 PRINTCHR$ (147),"      ARRIVEDERCI":E
ND
200 REM DATA PER DISEGNO
210 DATA207,183,183,183,183,183,208
220 DATA180,32,32,32,32,32,170
230 DATA180,113,32,32,32,32,170
240 DATA180,32,32,113,32,32,170
250 DATA180,32,32,32,32,113,170
260 DATA180,113,32,32,32,113,170
270 DATA204,175,175,175,175,175,186
300 REM DATA PER OGNI FACCIA DEL DADO
310 DATA0,1,1,3,1,1,6
320 DATA0,4,1,1,1,2,6
330 DATA0,4,1,3,1,2,6
340 DATA0,5,1,1,1,5,6
350 DATA0,5,1,3,1,5,6
360 DATA0,5,1,5,1,5,6
```

Altrettanto facilmente può essere simulato il lancio contemporaneo di due dadi. Il risultato possibile in ogni lancio è un numero compreso tra 2 e 12. Come certo saprete, alcuni punteggi si ripetono più facilmente di altri. Questo fatto risulterà chiaro con uno dei due programmi seguenti.

Se due dadi sono lanciati più volte, allora la frequenza (o probabilità) che si verifichi un certo risultato è data nella tabella seguente.

Punteggio	Probabilità
2	1/36
3	2/36
4	3/36
5	4/36
6	5/36
7	6/36
8	5/36
9	4/36
10	3/36
11	2/36
12	1/36

```
10 REM DUE DADI
20 PRINTCHR$ (147)," LANCIO DI DUE DADI"
CHR$ (17)
30 PRINT" IL PROGRAMMA SIMULA IL LANCIO
DI DUE    DADI NORMALI"CHR$ (17)
40 PRINT"PREMERE S PER CONTINUARE":K=0
50 GETG$:IFG$<>"S"THEN50
60 Y=RND(-TI)
70 K=K+1:PRINTCHR$ (147),"     PROVA NUME
RO"STR$(K)CHR$ (17)
80 FORI=1TO60
90 L=INT(RND(1)*6)+1:M=INT(RND(1)*6)+1:A
(L+M-2)=A(L+M-2)+1
100 PRINTCHR$ (154);"**";L;M;
110 NEXT
120 PRINT:PRINT"CONTEGGIO DEI RISULTATI"
130 FORI=0TO10
140 PRINTSTR$(I+2);"="A(I),:IFI=3ORI=7TH
ENPRINT
150 NEXT
160 PRINT:PRINT:PRINT:PRINTCHR$ (154)"AN
CORA? S/N"
```

```
170 GETG$:IFG$<>"S"ANDG$<>"N"THEN170
180 IFG$="S"THEN60
190 PRINTCHR$ (147)"     ARRIVEDERCI":EN
D


10 REM DUE DADI DISEGNATI
20 PRINTCHR$ (147)"     LANCIO DI DUE DAD
I CON DISEGNO"CHR$ (17)
30 PRINT" IL PROGRAMMA SIMULA IL LANCIO
DI DUE     DADI NORMALI"CHR$ (17)
40 FORI=0TO6:FORJ=0TO6:READA:A$(I)=A$(I)
+CHR$ (A):NEXT:NEXT
50 FORI=1TO6:FORJ=0TO6:READA:D$(I,J)=A$(
A):NEXT:NEXT
60 PRINT"PREMERE S PER CONTINUARE":K=0
70 GETG$:IFG$<>"S"THEN70
80 Y=RND(-TI)
90 K=K+1:PRINTCHR$ (147)"    LANCIO NUME
RO"STR$(K),CHR$ (17)
100 L=INT(RND(1)*6)+1:M=INT(RND(1)*6)+1:
A(L+M-2)=A(L+M-2)+1
110 FORJ=0TO7:PRINTTAB(10)D$(L,J);SPC(6)
D$(M,J):NEXT
120 PRINT:PRINT:PRINT"CONTEGGIO DEI RISU
LTATI";CHR$ (17)
130 FORI=0TO10
140 PRINTSTR$(I+2);"=";A(I),:IFI=3ORI=7T
HENPRINT
150 NEXT
160 PRINT:PRINT:PRINT:PRINTCHR$ (154)"AN
CORA? S/N"
170 GETG$:IFG$<>"S"ANDG$<>"N"THEN170
180 IFG$="S"THEN80
190 PRINTCHR$ (147)"     ARRIVEDERCI":EN
D
200 REM DATA PER DISEGNO
210 DATA207,183,183,183,183,183,208
220 DATA180,32,32,32,32,32,170
230 DATA180,113,32,32,32,32,170
240 DATA180,32,32,113,32,32,170
250 DATA180,32,32,32,32,113,170
```

```
260 DATA180,113,32,32,32,113,170
270 DATA204,175,175,175,175,175,186
300 REM DATA PER OGNI FACCIA DEL DADO
310 DATA0,1,1,3,1,1,6
320 DATA0,4,1,1,1,2,6
330 DATA0,4,1,3,1,2,6
340 DATA0,5,1,1,1,5,6
350 DATA0,5,1,3,1,5,6
360 DATA0,5,1,5,1,5,6
```

Il gioco delle carte

Un normale mazzo di carte ne contiene 52 e ognuna può egualmente apparire in cima al mazzo ben mescolato. Il programma che segue, Carte, illustra come il Commodore 64 può simulare la distribuzione di carte da un mazzo ben mescolato. Ad ogni nuova mano, si assume che la carta estratta precedentemente sia rimessa nel mazzo e questo venga rimescolato bene.

```
10 REM CARTE
20 POKE53281,15:PRINTCHR$ (147),CHR$ (31
)"       CARTE"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA SIMULA L'ESTRA
ZIONE"CHR$ (17)
40 PRINT"DI UNA CARTA DA UN MAZZO BEN ME
SCOLATO"CHR$ (17)
50 DIMA$(12):FORI=0TO12:READA$(I):NEXT
60 FORI=0TO3:READA:B$(I)=CHR$ (A):NEXT
70 FORI=0TO3:READA:C$(I)=CHR$ (A):NEXT
80 PRINT"          PREMERE S PER CONTINUAR
E":K=0
90 GETG$:IFG$<>"S"THEN90
100 Y=RND(-TI)
110 K=K+1:PRINTCHR$ (147)," ESTRAZIONE N
UMERO"STR$(K)CHR$ (17)
120 L=INT(RND(1)*13):M=INT(RND(1)*4)
130 PRINTTAB(18+(L=9))C$(M)A$(L)"  "B$(M
)
140 PRINT:PRINTCHR$ (31),"     ANCORA? S/
N"
150 GETG$:IFG$<>"S"ANDG$<>"N"THEN150
```

```
160 IFG$="S"THEN100
170 PRINTCHR$ (147),"      ARRIVEDERCI":E
ND
200 REM DATA
210 DATAA,2,3,4,5,6,7,8,9,10,J,Q,K
220 DATA120,122,115,97
230 DATA 144,28,28,144
```

Nel programma Carte,il calcolatore prima sceglie un numero tra 0 e 12 che (aggiungendo 1) determina il numero sulla faccia della carta, poi seleziona un numero tra 0 e 3 per ottenere il seme rappresentato. Come alternativa, si potrebbe scegliere un numero tra 0 e 51.

K = INT(RND(1)*52)

e ricavare il numero ed il seme della carta da questo numero, nel seguente modo

L = INT(K/4):M = K–4*L

L determina il numero e M il seme della carta.

Il programma Carte sceglie una carta dal mazzo mescolato e prima di ogni nuova scelta, la rimette nel mazzo e lo rimescola. Che succede se mescoliamo il mazzo una volta sola, all'inizio, e poi sfiliamo le carte nell'ordine in cui appaiono dalla cima del mazzo al fondo? È necessaria una routine diversa per ottenere questo. In pratica numeriamo le carte da 0 a 51 e poi rimescoliamo casualmente questi 52 numeri. Questo rimescolamento è fatto sistematicamente scambiando casualmente la Prima carta con una delle altre 51, poi la seconda con una delle rimanenti 50 e così via. Il programma 'Mescola le Carte' mostra questa tecnica.

```
10 REM MESCOLA LE CARTE
20 POKE 53281,15:PRINTCHR$ (147);CHR$ (3
1),"  MESCOLA LE CARTE"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA SIMULA IL MESC
OLAMENTO"
40 PRINT"DI UN MAZZO DI CARTE",CHR$ (17)
50 DIMA$(12):FORI=0TO12:READA$(I):NEXT
60 FORI=0TO3:READA:B$(I)=CHR$ (A):NEXT
70 FORI=0TO3:READA:C$(I)=CHR$ (A):NEXT
80 DIMD%(51):FORI=0TO51:D%(I)=I:NEXT
```

```
90 PRINT"          PREMERE S PER CONTINUAR
E"
100 GETG$:IFG$<>"S"THEN100
110 K=0:Y=RND(-TI)
120 K=K+1:PRINTCHR$ (147),"MESCOLAMENTO
NUMERO"STR$(K)CHR$ (17)
130 REM MESCOLAMENTO
140 FORI=0TO50
150 L=INT(RND(1)*(52-I))+I
160 REM L SODDISFA  I<=L<=N
170 T=D%(I):D%(I)=D%(L):D%(L)=T
180 NEXT
190 FORI=0TO51
200 L=INT(D%(I)/4):M=D%(I)-4*L:IFL<>9THE
NPRINT" ";
210 PRINTC$(M)A$(L)"  "B$(M),:
220 NEXT
230 PRINT:PRINT:PRINTCHR$ (31),"      ANC
ORA S/N":PRINT:PRINT:PRINT
240 GETG$:IFG$<>"S"ANDG$<>"N"THEN240
250 IFG$="S"THEN120
260 PRINTCHR$ (147),"      ARRIVEDERCI":E
ND
300 REM DATA
310 DATAA,2,3,4,5,6,7,8,9,10,J,Q,K
320 DATA120,122,115,97
330 DATA144,28,28,144
```

Eventi non equiprobabili

La maggior parte degli esempi che abbiamo visto ha la proprietà che uno qualsiasi degli eventi avviene con la stessa frequenza di ogni altro. L'esempio che segue invece no.

Un secchiello contiene 100 bottoni colorati: ce ne sono 6 rossi, 54 blu e 40 verdi. Per simulare l'estrazione di un bottone dal secchiello usiamo le seguenti istruzioni.

```
X = RND(1)
R$="ROSSO"
IF X >=0.06 THEN R$="BLU"
IF X >=0.60 THEN R$="VERDE"
PRINT R$
```

Il programmal Bottoni simula l'estrazione di un bottone dal secchiello. Vengono fatte 100 estrazioni, ma dopo ognuna il bottone è rimesso al suo posto.

```
10 REM BOTTONI
20 POKE 53281,15:PRINTCHR$ (147);CHR$ (1
54),"       BOTTONI"CHR$ (17)
30 PRINT"QUESTO PROGRAMMA ILLUSTRA L'EST
RAZIONE"
40 PRINT"DI UN BOTTONE DA UN SECCHIELLO
CHE NE"
50 PRINT"CONTIENE 6 ROSSI, 54 BLU E 40 V
ERDI."
55 PRINT"DOPO L'ESTRAZIONE, IL BOTTONE E
' RIMESSO";
57 PRINT"NEL SECCHIELLO"CHR$ (17)
60 PRINT"          PREMERE S PER CONTINUAR
E":K=0
70 GETG$:IFG$<>"S"THEN70
80 Y=RND(-TI)
90 K=K+1:PRINTCHR$ (147),"ESTRAZIONE NUM
ERO"STR$(K)CHR$ (17)
100 X=RND(1):IFX<0.06THENR$=CHR$ (28)+CH
R$ (166)+"  ROSSO"
110 IFX>=0.06THENR$=CHR$ (31)+CHR$ (166)
+"  BLU"
120 IFX>=0.6THENR$=CHR$ (30)+CHR$ (166)+
"  VERDE"
130 PRINTTAB(16)R$:PRINT:PRINTCHR$ (154)
"CONTEGGIO:  ";
140 S=ASC (R$)-28:A(S)=A(S)+1
150 PRINTCHR$ (31)"BLU";A(3),;
160 PRINTCHR$ (28)"ROSSI";A(0),;
170 PRINTCHR$ (30)"VERDI";A(2),;
180 PRINT:PRINT:PRINTCHR$ (31),"      ANC
ORA S/N":PRINT:PRINT:PRINT
190 GETG$:IFG$<>"S"ANDG$<>"N"THEN190
200 IFG$="S"THEN90
210 PRINTCHR$ (147),"      ARRIVEDERCI":P
OKE53291,6:END
```

CAPITOLO 14

DATI IN EVIDENZA

Che cosa fate quando vi trovate di fronte ad un grande numero di dati? Per esempio, qui c'è un gruppo di numeri che potrebbero essere il risultato di un qualche esperimento, o di un controllo etc.

```
23 67 89 45 40 10  5
19 99 40 23  9 11 21
34 34 56 41 42 27 80
```

Potete ottenere una grande quantità di informazioni su questi dati, osservando alcuni parametri indicativi quali il valore medio, la varianza e la deviazione standard. Ma prima, i dati devono essere inseriti nel vostro calcolatore.

Tali dati possono essere memorizzati nel vostro Commodore 64 in forma di tabelle X(I), cosicché X(0)=23, X(1)=67, ecc. e potrebbero essere introdotti in maniera interattiva, come è illustrato nei programmi Introduzione Dati I e Introduzione Dati II.

```
200 REM INTRODUZIONE DATI I
210 PRINTCHR$ (147),CHR$ (154),"INTRODUZ
IONE DATI",CHR$ (17)
220 PRINT" QUESTO VI PERMETTE DI INTRODU
RRE DATI  NUMERICI (ALMENO 2).",CHR$ (17
)
230 PRINT"QUANTI DATI VOLETE INTRODURRE?
",CHR$ (17)
240 INPUT"NUMERO ";N:PRINT
250 IFN<2ORN<>INT(N)THENPRINT"PENSACI BE
NE!":GOTO240
260 N=N-1:DIMX(N)
270 PRINT"INTRODUCI ORA I"N+1"DATI",CHR$
 (17)
280 FORI=0TON
290 PRINT"POSIZIONE";I+1;:INPUTX(I)
```

```
300 NEXT
310 PRINT:PRINT,"PREMI S PER CONTINUARE"
320 GETG$:IFG$<>"S"THEN320
```

Introduzione Dati I richiede, all'inizio, la quantità di dati da introdurre e la tabella X(I) è dimensionata di conseguenza. Al contrario, Introduzione Dati II assume che non saranno introdotti più di 100 numeri. Inserite il dato richiesto e per finire date -99999. Se necessario, il numero M alla linea 240 può essere modificato da 100 ad altro valore.

Questo capitolo contiene diversi brevi programmi che possono essere aggiunti ai programmi Introduzione Dati ed il risultato finale costituirà un programma che vi sarà utile per analizzare i vostri dati.

```
200 REM INTRODUZIONE DATI II
210 PRINTCHR$ (147)CHR$ (154)," INTRODU
ZIONE DATI",CHR$ (17)
220 PRINT"QUESTO PROGRAMMA VI PERMETTE D
I"
225 PRINT"INTRODURRE DATI  NUMERICI (ALM
ENO 2).",CHR$ (17)
230 PRINT"INTRODUCETE I DATI UNO PER VOL
TA",CHR$ (17)
240 M=100:DIMX(M)
250 FORI=0TOM
260 PRINT;:IFI>1THENPRINT,CHR$ (154)SPC(
(I+1)/10)"DATE -99999 PER FINIRE."
270 PRINTCHR$ (158)"POSIZIONE";I+1
280 INPUTX(I):IFX(I)=-99999ANDI>1THENN=I
-1:I=M
290 IFX(I)=-99999ANDI<2THENPRINTCHR$ (15
4)"TROPPO PRESTO PER FINIRE.":I=I-1
300 NEXT
310 PRINT:PRINT,"PREMI S PER CONTINUARE"
320 GETG$:IFG$<>"S"THEN320
```

La media

La *media* (o norma) è un importante dato statistico che è ottenuto sommando tra loro i dati in valore e dividendo per il numero totale dei dati.

$$\text{media} = \frac{\text{somma dei dati}}{\text{numero dei dati}}$$

Se i numeri sono memorizzati nella tabella X(I) con I tra 0 e N-1, allora XM può essere calcolato con le seguenti linee di programma:

```
400 REM MEDIA DEI DATI
410 PRINTCHR$ (147),CHR$ (154)," ANALISI
 DATI"CHR$ (17)CHR$ (158)
415 PRINT"NUMERO DATI =";N+1;CHR$ (17)
420 X=0:FORI=0TON:X=X+X(I):NEXT:XM=X/(N+
1)
430 PRINT"MEDIA =";XM;CHR$ (17)
```

Massimo, minimo e dispersione

È spesso utile conoscere il massimo ed il minimo valore dei dati e si può realizzare una piccola ricerca con il vostro Commodore 64. Ciò che segue è un programmino che realizza quanto richiesto e calcola, in più, l'*intervallo* o *dispersione* dei dati che è semplicemente la differenza tra il valore più grande e quello più piccolo.

```
430 PRINT"MEDIA =";XM;CHR$ (17)
500 REM MAX, MIN E DISPERSIONE DATI
510 MA=-1E38:MI=1E38
520 FORI=0TON
530 IFX(I)>MATHENMA=X(I)
540 IFX(I)<MITHENMI=X(I)
550 NEXT
560 PRINT"IL VALORE MIN E'  =";MI
570 PRINT"IL VALORE MAX E'  =";MA
580 PRINT"LA DISPERSIONE E' =";MA-MI;CHR
$ (17)
```

Volendo ottenere i dati ordinati per valori crescenti o decrescenti, si può utilizzare uno dei tanti sistemi esistenti (quali bubble sort, quick sort, shell sort etc.) (sort=ordinamento) i cui dettagli non vengono qui forniti.

Deviazione standard e varianza

La media è uno strumento semplice, utile e potente, ma non ci dice tutto quello che ci serve sapere. Ad esempio, osserviamo questo gruppo di dati:

DATI per X(I) 20, 21, 20, 19
DATI per Y(I) 38, 26, 14, 2

Il valore di XM e YM è sempre 20, però vi è una maggiore variazione nei dati Y(I) che in quelli di X(I) e questa può essere misurata utilizzando la deviazione standard.

La *deviazione standard* di un gruppo di dati è data dalla seguente formula.

$$\text{deviazione standard} = \text{SQR}\ \frac{\text{somma (differenza tra ogni valore e la media)}}{\text{numero dei dati} - 1}$$

La *varianza* è il quadrato della deviazione standard.

Il procedimento di calcolo per la deviazione standard di dati inserita in una tabella X(I) si svolge così:

1. Calcolate la media XM.
2. Trovate la deviazione della media, cioè il valore X(I) — XM.
3. Elevate al quadrato ogni deviazione, cioè trovate $(X(I)-XM)^2$
4. Sommate i quadrati delle deviazioni.
5. Dividete per il numero di termini — 1 per ottenere la varianza XV.
6. Fate la radice quadrata: questa è la deviazione standard XD dei dati.

La deviazione standard fornisce un'idea di quanto i dati sono sparsi o disseminati nell'intorno del valore medio.

Prendiamo ancora l'esempio:

DATI per X(I) 20, 21, 20, 19
DATI per Y(I) 38, 26, 14, 2

La deviazione standard è data dai seguenti calcoli.

```
XD = SQR((0*0 + 1*1 + 0*0 + (-1)*(-1))/3
   = SQR(2/3)
   = 0.816496581
```

$$YD = SQR((18*18+6*6+(-6)*(-6)+(-18)*(-18))/3$$
$$= SQR(240)$$
$$= 15.4919334$$

Questo effettivamente riflette la differenza nella dispersione dei dati.

Il programma che segue, calcola la deviazione standard XD dei dati memorizzati in una tabella X(I).

```
600 REM DEVIAZIONE STANDARD"
610 X=0:FORI=0TON:Y=X(I)-XM:X=X+Y*Y:NEXT
620 XD=SQR(X/N)
630 PRINT"DEVIAZIONE STANDARD =";XD;CHR$
 (17)
```

Intervalli di confidenza

La deviazione standard è utile perché indica su quale estensione sono distribuiti i dati intorno al valore medio. In molti processi produttivi, il prodotto varia leggermente per dimensione o per qualità o per lunghezza etc. Ci riferiamo agli elementi che stiamo misurando come ad una popolazione. La variazione in una popolazione è sovente distribuita normalmente.

Gli statisti hanno scoperto che la curva normale o distribuzione normale approssima un grande numero di casi reali. Se l'insieme di dati è grande (almeno più di 30) si assume spesso, per i calcoli, che la popolazione sia distribuita normalmente, anche se ciò potrebbe non essere del tutto vero.

Parlando in termini più semplici, una popolazione è normalmente distribuita se è simmetricamente dispersa intorno al valore medio: cioè con la maggioranza della popolazione vicinissima al valore medio e la minoranza distribuita lontano, su ambo i lati.

Più precisamente, in una popolazione distribuita normalmente, circa il 68% dei membri giace entro una deviazione standard dalla media e circa il 96% giace entro 2 deviazioni standard. Più in generale, la tabella qui sotto elenca le percentuali associate con i vari multipli della deviazione standard.

% di popolazione	Multipli della dev. standard
50	0.6745
68.27	1
80	1.28
90	1.645
95	1.96
95.45	2
99	2.575
99.73	3

La suddetta tabella mostra che ci dovremmo attendere che il 95% della popolazione si trovi entro 1.96 volte la deviazione standard dal valore medio. In altre parole il 95% della popolazione giace nell'intervallo

da XM − 1.96*XD a XM + 1.96*XD

dove XM è il valore medio e XD la deviazione standard. Ci riferiamo a questo intervallo dicendo che è l'intervallo di confidenza del 95% per quella popolazione. Analogamente l'intervallo di confidenza del 99% è

da XM − 2.575*XD a XM + 2.575*XD

Ecco un esempio che dimostra l'utilità degli intervalli di confidenza. Voi sospettate che un commerciante venda bottiglie di 2 litri di limonata non completamente riempite. Comprate 10 bottiglie e misurate attentamente il contenuto con i seguenti risultati (in litri):

2.001, 2.040, 2.020, 2.000, 2.015
2.006, 2.005, 2.031, 2.008, 2.018

Tutte le bottiglie contengono più o meno due litri: calcoleremo però la media e la deviazione standard di questi dati. I risultati sono:

XM = 2.0144
XD = 0.0132

Ora, assumendo che i nostri campioni provengano da una popolazione distribuita normalmente, possiamo calcolare gli intervalli di confidenza: quello del 95% nel nostro caso varia

da 2.0144 − 1.96*0.0132 a 2.0144 + 1.96*0.0132

cioè

da 1.989 a 2.040

Perciò ci aspettiamo che il 95% delle bottiglie contenga tra 1.989 e 2.040 litri. Ciò significa anche che il 2.5% delle bottiglie dovrebbe contenere più di 2.040 litri e il 2.5% dovrebbe contenere meno di 1.989 litri. Possiamo anche trovare l'intervallo di confidenza del 90% che è tra

2.0144 − 1.645*0.0132 e 2.0144 − 1.645*0.0132

ovvero tra

1.993 e 2.036

perciò si suppone che il 90% delle bottiglie contenga tra 1.993 e 2.036 litri. Ne consegue che il 5% delle bottiglie di limonata conterrà meno dei due litri richiesti. (Egualmente il 5% conterrà più di 2.036 litri).

Nell'esempio visto, si sono fatte molte ipotesi e la conclusione cui si è giunti non è sufficiente per portare avanti una causa legale.

La media della popolazione è stata calcolata da un campione. Come sappiamo che questa è la vera media della popolazione? La media può variare con il campione usato ma noi possiamo valutare quanto dista la media del campione dalla media reale utilizzando la deviazione standard. Possiamo dire con un intervallo di confidenza del 95% che la media varia tra

XM − 1.96*XD/SQR(N−1) e XM + 1.96*XD/SQR(N−1)

essendo XM la media calcolata sul campione di ampiezza N. Questo viene detto l'intervallo di confidenza del 95% della media. L'intervallo del 99% è compreso tra

XM−2.575*XD/SQR(N−1) e XM+2.575*XD/SQR(N−1)

In senso stretto, questi calcoli sono validi se l'ampiezza del campione è grande (diciamo maggiore di 30).

Non prendete troppo seriamente gli intervalli di confidenza e non confondete i due tipi tra loro.

Il programma che segue calcola gli intervalli di confidenza del 95% e del 99% della popolazione e della media.

```
700 REM INTERVALLI DI CONFIDENZA
710 C1=1.96*XD:M1=C1/SQR(N):C2=2.575*XD:
M2=C2/SQR(N)
720 PRINT"INTERVALLO DI CONFIDENZA DEL 9
5%:"
730 PRINT"POPOLAZIONE DA ";XM-C1;"A";XM+
C1
740 PRINT"VAL. MEDIO  DA ";XM-M1;"A";XM+
M1;CHR$ (17)
750 PRINT"INTERVALLO DI CONFIDENZA DEL 9
9%:"
760 PRINT"POPOLAZIONE DA ";XM-C2;"A";XM+
C2
770 PRINT"VAL. MEDIO  DA ";XM-M2;"A";XM+
M2;CHR$ (17)
```

Un tocco finale

I programmi di questo capitolo possono essere riuniti tra loro per ottenere un utile programma di analisi dei dati. In questo che segue sono date le necessarie parti di congiunzione.

```
10 REM ANALISI DATI
20 PRINTCHR$ (147)CHR$ (154),"   ANALISI
 DATI",CHR$ (17)
30 PRINT"QUESTO PROGRAMMA VI PERMETTE DI
 INTRODURRE DATI E ANALIZZARLI",CHR$ (17
)
40 PRINT"DOPO AVER INTRODOTTO I DATI, VI
 VERRANNO";
50 PRINT"FORNITE LE SEGUENTI INFORMAZION
I:",CHR$ (17)
60 PRINT"1. VALORE MEDIO DEI DATI",CHR$
(17)
70 PRINT"2. VALORE MAX, MIN E DISPERSION
E"CHR$ (17)
80 PRINT"3. DEVIAZIONE STANDARD"CHR$ (17
)
90 PRINT"4. INTERVALLI DI CONFIDENZA DEL
 95%,99%"CHR$ (17)
```

```
100 PRINT:PRINTCHR$ (154);"PREMERE S PER
 CONTINUARE"
110 GETG$:IFG$<>"S"THEN110
200 REM INTRODUZIONE DATI II
210 PRINTCHR$ (147)CHR$ (154),"   INTROD
UZIONE DATI",CHR$ (17)
220 PRINT"QUESTO PROGRAMMA VI PERMETTE D
I"
225 PRINT"INTRODURRE DATI  NUMERICI (ALM
ENO 2)."CHR$ (17)
230 PRINT"INTRODUCETE I DATI UNO PER VOL
TA",CHR$ (17)
240 M=100:DIMX(M)
250 FORI=0TOM
260 PRINT;:IFI>1THENPRINT,CHR$ (154)SPC(
(I+1)/10)"DATE -99999 PER FINIRE."
270 PRINT"POSIZIONE";I+1
280 INPUTX(I):IFX(I)=-99999ANDI>1THENN=I
-1:I=M
290 IFX(I)=-99999ANDI<2THENPRINTCHR$ (15
4)"TROPPO PRESTO PER FINIRE.":I=I-1
300 NEXT
310 PRINT:PRINT,"PREMERE S PER CONTINUAR
E"
320 GETG$:IFG$<>"S"THEN320
400 REM MEDIA DEI DATI
410 PRINTCHR$ (147),CHR$ (154)," ANALISI
 DATI"CHR$ (17)
415 PRINT"NUMERO DATI =";N+1;CHR$ (17)
420 X=0:FORI=0TON:X=X+X(I):NEXT:XM=X/(N+
1)
430 PRINT"MEDIA =";XM;CHR$ (17)
500 REM MAX, MIN E DISPERSIONE DATI
510 MA=-1E38:MI=1E38
520 FORI=0TON
530 IFX(I)>MATHENMA=X(I)
540 IFX(I)<MITHENMI=X(I)
550 NEXT
560 PRINT"IL VALORE MIN E'  =";MI
570 PRINT"IL VALORE MAX E'  =";MA
580 PRINT"LA DISPERSIONE E' =";MA-MI;CHR
$ (17)
```

```
600 REM DEVIAZIONE STANDARD"
610 X=0:FORI=0TON:Y=X(I)-XM:X=X+Y*Y:NEXT
620 XD=SQR(X/N)
630 PRINT"DEVIAZIONE STANDARD =";XD;CHR$
 (17)
700 REM INTERVALLI DI CONFIDENZA
710 C1=1.96*XD:M1=C1/SQR(N):C2=2.575*XD:
M2=C2/SQR(N)
720 PRINT"INTERVALLO DI CONFIDENZA DEL 9
5%:"
730 PRINT"POPOLAZIONE DA ";XM-C1;"A";XM+
C1
740 PRINT"VAL. MEDIO  DA ";XM-M1;"A";XM+
M1;CHR$ (17)
750 PRINT"INTERVALLO DI CONFIDENZA DEL 9
9%:"
760 PRINT"POPOLAZIONE DA ";XM-C2;"A";XM+
C2
770 PRINT"VAL. MEDIO  DA ";XM-M2;"A";XM+
M2
900 REM ENDING
910 PRINT"ANCORA? S/N"
920 GETG$:IFG$<>"S"ANDG$<>"N"THEN920
930 IFG$="S"THENRUN
940 PRINTCHR$ (147)CHR$ (154)"ARRIVEDERC
I":END
```

SOMMARIO

(Per capitoli)

1. Funzioni semplici

Visualizzazione ordinata dei numeri
Giustificazioni a destra di numeri
INT, la parte intera di un numero
ABS, il valore assoluto di un numero
Arrotondamento di numeri
Arrotondamento in basso e in alto
Arrotondamento di un numero a D cifre decimali
Saldi bancari
Saldi bancari negativi
SGN, il segno o funzione segno
Visualizzazione a colori dei saldi bancari

2. Trigonometria

Disegni in scala
Stima di altezze e distanze
Triangoli rettangoli
Funzioni trigonometriche, TAN, SIN e COS
Ipotenusa, lato adiacente, lato opposto
Radianti
Pi-greco
Da gradi a radianti e viceversa
Ricerca delle lunghezze dei lati di un triangolo rettangolo
Funzioni inverse
Arcotangente, ATN
Arcoseno, ASN
Arcocoseno, ACS
Teorema di Pitagora
Triangoli non rettangoli

Legge dei coseni
Legge dei seni
Ricerca dei lati e/o angoli di un triangolo
Rifrazione
Angolo d'incidenza, angolo di rifrazione
Legge di Snell
Indice di rifrazione
Riflessione
Angolo critico

3. Trigonometria Terrestre

La Terra
Linee rette sulla Terra
Cerchi massimi
Linee di longitudine
Greenwich, Inghilterra
Linee di Latitudine
Calcolo delle distanze tra due punti sulla Terra

4. Potenze

Quadrati di numeri
Potenze di numeri
Proprietà delle potenze
SQR, la Radice quadrata
Numeri immaginari
Numeri complessi
Numeri complessi sul Commodore 64
Equazioni quadratiche
Soluzioni di equazioni quadratiche
Radici, ricerca di radici
Formula per le radici dell'equazione quadratica
Discriminante
Soluzione di equazioni quadratiche sul 64
Soluzioni di altre equazioni
Equazioni polinomiali
Grado di un polinomio
Radici di un polinomio
Metodo di Newton
Derivata di un polinomio

Ricerca delle radici con il metodo Newton
EXP, funzione esponenziale
E, EXP(1)
Fattoriale
Proprietà delle funzioni esponenziali
Formula per le funzioni esponenziali
Funzione logaritmica
LOG, la funzione logaritmo naturale
Proprietà delle funzioni logaritmiche
Ricerca di radici di altre funzioni

5. Serie

Serie
Termini di una serie
Generazione di serie
Serie aritmetica, progressione aritmetica
Differenza comune
Generazione di sequenze aritmetiche
Quale preferite?
Serie geometrica, progressione geometrica
Rapporto comune
Generazione di serie geometriche
Interesse, interesse composto
Interesse giornaliero
Gioco di Lascia o raddoppia
Serie di Fibonacci
Generazione di serie di Fibonacci

6. Basi numeriche

Sistema decimale
Cifre
Rappresentazione decimale
Base
Coefficienti di un numero N in base B
Sistema numerico binario
Numeri esadecimali
Conversione di numeri da una base all'altra
Numeri sul Commodore 64
PEEK dei numeri sul 64

Forma binaria di numeri tra 0 e 1
Visualizzazione di numeri binari tra 0 e 1
Virgola mobile
Esponente binario
Mantissa binaria
1*(1+2↑(−24))=1+2↑(−25) secondo il 64

7. Giorni e settimane

Giorni della settimana
Congruenza di Zeller
Ricerca del giorno della settimana per una qualsiasi data
Visualizzazione di un calendario mensile per ogni mese di ogni anno
Gestione delle date
Data pseudo-giuliana
Elenco di date che distano un certo numero di giorni

8. Massimo Comun Divisore

Divisori comuni, fattori comuni
Massimo comun divisore, fattore comune maggiore
Algoritmo di Euclide
Calcolo del massimo comun divisore
Minimo comune multiplo (MCM)
Calcolo del minimo comune multiplo

9. Numeri Primi

Numeri primi
Numeri composti
Esistono infiniti numeri primi
Crivello di Eratostene
Verifica dei numeri primi
Ricerca dei fattori
Numeri primi grandi
Numeri di Mersenne, numeri primi di Mersenne
Il più grande numero primo conosciuto
Test di primalità probabilistica
Piccolo Teorema di Fermat
Numeri pseudo-primi in una qualsiasi base
La maggior parte di numeri pseudo-primi sono numeri primi

10. Miscellanea

Terne pitagoriche
Terne pitagoriche primitive
Generazione di terne pitagoriche primitive
Potenze in multiprecisione
Esatto calcolo di prodotti di grandi numeri con il 64
Esatto calcolo di potenze di numeri arbitrariamente grandi

11. Matrici

Matrici, tabelle rettangolari di numeri
Matrice M per N
Matrice quadrata
Somma di matrici
Perché sommare le matrici
Prodotto di matrici
Perché moltiplicare le matrici
Matrice nulla
Matrice identità
Inverso di una matrice, reciproco di una matrice
Calcolo dell'inverso di una matrice
Sistemi di equazioni
Soluzione di sistemi di equazioni

12. Codici

Crittografia
Cifre
Codici a sostituzione
Programma-tipo per codici a sostituzione
Codici a matrice
Uso di matrici per cifrare messaggi
Programma-tipo per codici a matrice
Codici a chiave pubblica
Numeri primi e codici sicuri

13. Casualità!

Testa o croce
Lancio di una moneta
Numeri casuali

Simulazione del lancio della moneta con il 64
Uomini e dadi
Lancio di un dado
Probabilità
Simulazione del lancio di un dado
Simulazione del lancio di due dadi
Giochi delle carte
Simulazione dell'estrazione di una carta
Rimescolamento casuale di un mazzo di carte
Eventi non egualmente probabili
Secchiello con 100 bottoni colorati
Simulazione dell'estrazione di un bottone dal secchiello

14. Dati in evidenza

Introduzione di dati numerici nel Commodore 64
Media, valore medio
Calcolo della media
Massimo e minimo
Dispersione, intervallo
Calcolo del massimo, minimo e dispersione
Deviazione standard
Varianza
Calcolo della deviazione standard
Intervalli di confidenza
Distribuzione normale
Calcolo di intervalli di confidenza per una popolazione
Calcolo di intervalli di confidenza per la media

SUNSHINE

Questo libro è scritto per tutti quelli che posseggono un Commodore 64 e vorrebbero conoscere qualche cosa di più riguardo alle tecniche matematiche.
Tutte le funzioni matematiche disponibili sul C 64 sono qui descritte, ed il loro uso è illustrato con programmi che possono essere utilizzati dal lettore all'interno dei suoi, per particolari applicazioni.
Il libro contiene anche informazioni e programmi su altri argomenti, come i codici e la crittografia, i numeri casuali, le serie, la trigonometria, i numeri primi e l'analisi statistica dei dati.

L. 24.000

Cod. 570D 88-7056-162-3